aws

Benutzer-Leitfaden

Amazon Inspector



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Inspector: Benutzer-Leitfaden

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon Inspector? ... 1
Features ... 1
Zugreifen auf Amazon Inspector ... 3
Erste Schritte ... 5
Vor der Aktivierung von Amazon Inspector ... 5
Tutorial „Erste Schritte“: Amazon Inspector aktivieren ... 6
Automatisierte Scans ... 12
Übersicht der Amazon Inspector-Scantypen ... 12
Einen Scantyp aktivieren ... 14
Scans aktivieren ... 15
Scannen Amazon EC2 EC2-Instances ... 16
Agent-based Scannen ... 17
Scannen ohne Agenten ... 23
Den Scanmodus verwalten ... 24
Instances von Amazon Inspector-Scans ausschließen ... 25
Unterstützte Betriebssysteme ... 26
Gründliche Inspektion für Linux-Instances ... 26
EC2-Instance wird Windows gescannt ... 32
Scannen von Amazon ECR-Container-Bildern ... 35
Scanverhalten für Amazon ECR-Scans ... 36
Zuordnung von Container-Images zu laufenden Containern ... 37
Unterstützte Betriebssysteme und Medientypen ... 38
Konfiguration der Dauer des erneuten Scans von Amazon ECR ... 39
Scannen mit Lambda-Funktionen ... 42
Scanverhalten beim Scannen mit Lambda-Funktionen ... 43
Unterstützte Laufzeiten und Funktionen ... 44
Standard-Scanning mit Amazon Inspector Lambda ... 44
Scannen von Lambda-Code mit Amazon Inspector ... 46
Deaktivierung eines Scan-Typs ... 48
Scans deaktivieren ... 49
CIS-Scans ... 51
Amazon EC2 EC2-Instance-Anforderungen für Amazon Inspector CIS-Scans ... 52
Amazon Virtual Private Cloud Cloud-Endpunktanforderungen für die Ausführung von CIS-Scans auf privaten Amazon EC2 EC2-Instances ... 53

CIS-Scans ausführen ... 54
Überlegungen zur Verwaltung von Amazon Inspector CIS-Scans mit AWS Organizations ... 55
Amazon Inspector-eigene Amazon S3 S3-Buckets, die für Amazon Inspector CIS-Scans verwendet werden ... 56
Eine CIS-Scankonfiguration erstellen ... 58
CIS-Scanergebnisse anzeigen ... 59
Bearbeitung einer CIS-Scankonfiguration ... 60
CIS-Scanergebnisse herunterladen ... 61
Amazon Inspector Codesicherheit ... 62
Voraussetzungen ... 62
Code-Sicherheit aktivieren ... 62
Einen vom Kunden verwalteten Zugriffsschlüssel erstellen AWS KMS ... 62
Eine Integration erstellen ... 65
Eine Integration erstellen für GitHub ... 66
Eine Integration erstellen für GitLab Self Managed ... 68
Anzeigen von Integrationen ... 71
Code-Repositorys anzeigen ... 71
Eine Integration löschen ... 73
Eine Scan-Konfiguration erstellen ... 73
Scan-Konfigurationen anzeigen ... 76
Bearbeitung einer Scan-Konfiguration ... 77
Löschen einer Scan-Konfiguration ... 78
Einen Scan auf Anforderung durchführen ... 78
Unterstützte Sprachen ... 79
Deaktivierung der Codesicherheit ... 80
Grundlegendes zu Erkenntnissen ... 81
Erkenntnistypen ... 82
Sicherheitslücke im Package ... 82
Sicherheitslücke im Code ... 83
Erreichbarkeit über das Netzwerk ... 84
Ergebnisse anzeigen ... 85
Anzeigen von Ergebnisdetails ... 86
Den Amazon Inspector Score anzeigen ... 90
Amazon Inspector-Punktzahl ... 90
Informationen zu Sicherheitslücken ... 92
Grundlegendes zu den Schweregraden der Ergebnisse ... 93

Schweregrad der Sicherheitslücke im Softwar ... 93
Schweregrad der Sicherheitslücke ... 95
Schweregrad der Erreichbarkeit des Netzwerks ... 93
Verwaltung der Erkenntnisse ... 97
Filtern von Ergebnissen ... 97
Filter in der Amazon Inspector Inspector-Konsole erstellen ... 97
Unterdrücken von Ergebnissen ... 98
Eine Unterdrückungsregel erstellen ... 99
Unterdrückte Ergebnisse anzeigen ... 100
Eine Unterdrückungsregel bearbeiten ... 100
Löschen einer Unterdrückungsregel ... 100
Ergebnisberichte exportieren ... 101
Schritt 1: Überprüfen Sie Ihre Berechtigungen ... 102
Schritt 2: Konfigurieren Sie einen S3-Bucket ... 104
Schritt 3: Konfigurieren Sie eine AWS KMS key ... 108
Schritt 4: Konfigurieren und exportieren Sie einen Ergebnisbericht ... 111
Beheben von Fehlern ... 114
Automatisieren von Reaktionen auf Ergebnisse mit EventBridge ... 115
Ereignisschema ... 116
Erstellung einer EventBridge Regel, um Sie über Ergebnisse von Amazon Inspector zu informieren ... 118
EventBridge für Amazon Inspector-Umgebungen mit mehreren Konten ... 123
Dashboard ... 124
Das Dashboard anzeigen ... 124
Dashboard-Komponenten verstehen ... 125
Die Schwachstellen-Datenbank durchsuchen ... 129
Die Schwachstellen-Datenbank durchsuchen ... 129
CVE-Details verstehen ... 130
CVE-Details ... 130
Informationen zu Sicherheitslücken ... 130
Referenzen ... 131
Exportieren SBOMs ... 132
Amazon Inspector Inspector-Formate ... 132
Filtert für SBOMs ... 138
Konfigurieren und exportieren SBOMs ... 138
EventBridge Schema ... 141

EventBridge Amazon-Basisschema für Amazon Inspector 141
Beispiel für das Auffinden von Ereignissen in Amazon Inspector 142
Beispiel für ein vollständiges Amazon Inspector Inspector-Ereignisschema für den ersten Scan 164
Beispiel für ein Amazon Inspector Inspector-Abdeckungsereignisschema 166
Beispiel für ein Schema zur auto Aktivierung von Amazon Inspector 168
Amazon Inspector VM-Scanner 170
-Übersicht 170
Vorteile 170
Gründliche Inspektion 171
Inspector VM Scanner aktivieren 172
Voraussetzungen 172
Enhanced EC2 Scanning aktivieren 172
Amazon VPC-Endpunktanforderungen für erweitertes EC2-Scannen auf privaten Amazon EC2 EC2-Instances 173
Häufig gestellte Fragen 174
Third-party Softwarezuweisung 175
Manuelle Installation und Konfiguration 175
Manuelles Installieren und Deinstallieren von Inspector VM Scanner 176
Installationspfade 180
Deinstallation von Inspector VM Scanner 180
Inspector VM Scanner ausführen 180
Ausgabe anzeigen 180
Anzeigen von Protokollen 181
Logs mit Systemd anzeigen 181
Fortgeschrittene Konfiguration 182
Konfiguration lokaler Ausgaben 182
Konfiguration der Ressourcennutzung 183
Scanziele konfigurieren 184
Verwaltung der regelmäßigen Ausführung 185
SSM-Plugin 188
Das Amazon Inspector SSM-Plugin für Linux 188
Das Amazon Inspector SSM-Plugin für Windows 189
Azure-Integration 191
Funktionsweise 191
Wie AWS Config wird verwendet 191

Arten von Anschlüssen ... 191
Methoden zum Scannen ... 192
Voraussetzungen ... 192
Azure konfigurieren ... 195
Amazon Inspector konfigurieren ... 200
Löschen eines Connectors ... 212
Amazon Inspector SBOM-Generator ... 215
Unterstützte Pakettypen ... 215
Unterstützte Konfigurationsprüfungen für Container-Images ... 216
Installation Sbomgen ... 216
Verwenden Sbomgen ... 218
Generieren Sie eine SBOM für ein Container-Image und geben Sie das Ergebnis aus ... 218
Generieren Sie eine SBOM aus Verzeichnissen und Archiven ... 219
Generieren Sie eine SBOM aus Go oder Rust kompilierte Binärdateien ... 220
Generieren Sie eine SBOM aus bereitgestellten Volumes ... 220
Senden Sie eine SBOM zur Identifizierung von Sicherheitslücken an Amazon Inspector ... 221
Verwenden Sie zusätzliche Scanner, um die Erkennungsfunktionen zu verbessern ... 223
Optimieren Sie Containerscans, indem Sie die maximale zu scannende Dateigröße anpassen ... 224
Beschränken Sie die Abrufgröße für Container-Images ... 225
Erkennt Scratch-Bilder mit Dateisystem-Heuristiken ... 226
Sammeln Sie Metadaten für laufende Prozesse ... 227
Evaluieren Sie CIS Docker Benchmark-Checks ... 228
Zeichnen Sie die FIPS-Konfigurationsindikatoren auf ... 229
Deaktivieren Sie die Fortschrittsanzeige ... 230
Authentifizierung bei privaten Registern mit Sbomgen ... 230
Authentifizieren Sie sich mit zwischengespeicherten Anmeldeinformationen (empfohlen) ... 231
Authentifizieren Sie sich mit der interaktiven Methode ... 231
Authentifizieren Sie sich mit der nicht interaktiven Methode ... 231
Beispielausgaben von Sbomgen ... 232
Frühere Versionen ... 234
Erfassung von Betriebssystemen ... 252
Unterstützte Betriebssystemartefakte ... 253
APK-based Sammlung von OS-Paketen ... 254
DPKG-based Sammlung von Betriebssystempaketen ... 255
RPM-based Sammlung von Betriebssystempaketen ... 256

Sammlung der Windows-Betriebssystemversionen ... 258
Sammlung von Chainguard-Image-Paketen ... 258
Sammlung von Image-Paketen ohne Distribution ... 259
MiniMOS-Paketsammlung ... 261
Gehärtete Docker-Images Sammlung von Paketen ... 261
Gehärtete Red Hat-Images Sammlung von Paketen ... 263
CleanStart Sammlung von Paketen mit gehärteten Bildern ... 264
Erfassung von Abhängigkeiten ... 265
Gehen Sie zum Scannen von Abhängigkeiten ... 265
Scannen von Java-Abhängigkeiten ... 269
JavaScript Scannen von Abhängigkeiten ... 273
.NET-Abhängigkeiten scannen ... 280
Scannen von PHP-Abhängigkeiten ... 285
Scannen von Python-Abhängigkeiten ... 288
Scannen von Abhängigkeiten in Ruby ... 295
Scannen von Abhängigkeiten auf Rost ... 298
Schnelles Scannen von Abhängigkeiten ... 302
Nicht unterstützte Artefakte ... 305
Sammlung von Ökosystemen ... 306
Unterstützte Ökosysteme ... 307
7-Zip Sammlung von Ökosystemen ... 312
Fähigkeiten von KI-Agenten Sammlung von Ökosystemen ... 313
AI-IDE-Erweiterungen Sammlung von Ökosystemen ... 314
Amazon Q Developer Sammlung von Ökosystemen ... 315
Apache Sammlung von Ökosystemen ... 316
Atlassian Sammlung von Ökosystemen ... 322
Claude Code Sammlung von Ökosystemen ... 325
Conda Sammlung von Ökosystemen ... 326
Curl Sammlung von Ökosystemen ... 327
Cursor Sammlung von Ökosystemen ... 330
Docker-Binärdateien Sammlung von Ökosystemen ... 331
Elasticsearch Sammlung von Ökosystemen ... 332
Generische AI/ML Modelle Sammlung von Ökosystemen ... 333
GitHub Copilot Sammlung von Ökosystemen ... 334
Google Sammlung von Ökosystemen ... 335
Homebrew Sammlung von Ökosystemen ... 337

HuggingFace Sammlung von Ökosystemen ... 338
Java Sammlung von Ökosystemen ... 339
Jenkins Sammlung von Ökosystemen ... 341
Kiro CLI Sammlung von Ökosystemen ... 343
MariaDB und MySQL Sammlung von Ökosystemen ... 344
Microsoft-Anwendungen Sammlung von Ökosystemen ... 346
Microsoft SQL Server Sammlung von Ökosystemen ... 351
MongoDB Sammlung von Ökosystemen ... 352
Mozilla Sammlung von Ökosystemen ... 353
Nginx Sammlung von Ökosystemen ... 356
Node.JS Runtime-Sammlung ... 357
Ollama Sammlung von Ökosystemen ... 358
Ollama Model Collector Sammlung von Ökosystemen ... 360
Sammlung des OpenSSH-Ökosystems ... 360
Sammlung des OpenSSL-Ökosystems ... 362
Oracle Database Server-Sammlung ... 363
PHP Sammlung von Ökosystemen ... 364
Redis Sammlung von Ökosystemen ... 365
Sammlung eigenständiger Ökosysteme für KI-Anwendungen ... 366
Windsurfen Sammlung von Ökosystemen ... 367
WordPress Sammlung von Ökosystemen ... 368
Zed Sammlung von Ökosystemen ... 371
SSL/TLS Scans von Zertifikaten ... 372
Verwenden Sbomgen Scans von Zertifikaten ... 372
Sammlung von Lizenzen ... 376
Sammeln Sie Lizenzinformationen ... 376
Unterstützte Pakete ... 377
Paket-URLs ... 383
PURL-Struktur ... 383
Versionsreferenzen ... 386
Empfehlungen ... 386
Java ... 386
JavaScript ... 387
Python ... 387
Verwenden von Namespaces CycloneDX ... 388
amazon:inspector:sbom_scanner-Namespace-Taxonomie ... 388

amazon:inspector:sbom_generator-Namespace-Taxonomie ... 390
Leitfaden für Plugin-Entwickler ... 397
-Übersicht ... 397
Schnellstart: Neue Plugins erstellen ... 398
IDE-Setup ... 400
Struktur des Plugin-Verzeichnisses ... 400
Discovery-Plugins ... 402
Plugins für die Sammlung ... 407
Ihre Plugins ausführen ... 411
Verfügbare Scanner auflisten ... 412
Auswahl des Scanners ... 413
Kollisionsregeln für Plug-ins ... 417
Debuggen ... 418
API-Referenz ... 421
Fehlerbehandlung ... 421
Sandbox-Einschränkungen ... 421
Code zwischen Plugins teilen ... 423
Leitfaden zum Testen von Plugins ... 424
Schnellstart ... 424
Verzeichnis-Layout ... 425
Das Testen.* API ... 426
Ausführen von Tests ... 428
Plugin-Helfer testen ... 429
Verhalten und Invarianten ... 430
Einschränkungen ... 431
Verantwortlichkeiten des Entwicklers ... 431
Plugin-API-Referenz ... 432
-Übersicht ... 432
Globaler Lebenszyklus von Plugins ... 434
Datei I/O ... 439
FileHandle Methoden ... 446
Binäre Dienstprogramme ... 447
Paket-Ausgabe ... 451
Herkunft des Package ... 455
Komponenten-Hashes ... 457
Komponentenlizenzen ... 459

Eigenschaften von CyclonedX ... 461
Eigenschaftskonstanten ... 463
Scanner-Gruppen ... 466
Konstanten für den Komponententyp ... 467
Konstanten vom Typ PURL ... 468
Plattformkonstanten ... 468
Konstanten für den Hash-Algorithmus ... 469
Informationen zum Artifact ... 470
Informationen zum System ... 473
Reguläre Ausdrücke ... 473
Strukturiertes Parsen ... 475
Windows Registry ... 476
Protokollierung ... 477
Debugging-Funktionen ... 478
API testen ... 478
Fehlerbehandlung ... 481
CI/CD Integration ... 483
Plugin-Integration ... 483
Unterstützte CI/CD Lösungen ... 484
Maßgeschneiderte Integration ... 485
Richten Sie ein Konto für die CI/CD Integration ein ... 485
Melde dich für eine an AWS-Konto ... 486
Konfigurieren Sie eine IAM-Rolle für die Integration CI/CD ... 486
Amazon Inspector Dockerfile-Prüfungen ... 488
Verwenden Sbomgen Dockerfile-Prüfungen ... 488
Unterstützte Dockerfile-Prüfungen ... 490
Eine benutzerdefinierte CI/CD Integration erstellen ... 496
Schritt 1. Konfigurieren AWS-Konto ... 496
Schritt 2. Installation Sbomgen Binary ... 496
Schritt 3. Verwenden Sbomgen ... 496
Schritt 4. Aufrufen der Amazon Inspector Scan API ... 497
(Optional) Schritt 5. Generieren und scannen Sie SBOM mit einem einzigen Befehl ... 497
API-Ausgabeformate ... 498
GitHub-Aktionen ... 510
Jenkins-Plugin ... 510
Schritt 1. Richten Sie eine ein AWS-Konto ... 511

Schritt 2. Installieren Sie den Amazon Inspector Jenkins Plugin 511
(Optional) Schritt 3. Addition Docker Anmeldeinformationen für Jenkins 511
(Optional) Schritt 4. Addition AWS Anmeldeinformationen für Jenkins 512
Schritt 5. Sobald Sie die Details auf dieser Seite überprüft haben, klicken Sie auf Jenkins um den HTML-Bericht anzuzeigen 514
Schritt 6: Fügen Sie Amazon Inspector Scan zu Ihrem Build hinzu 515
Schritt 7. Sehen Sie sich Ihren Amazon Inspector Inspector-Schwachstellenbericht an 521
Fehlerbehebung 522
CodeCatalyst-Aktionen 525
CodePipeline-Aktionen 526
TeamCity-Plugin 526
GitLab-Komponenten 526
Bewerten der Abdeckung 528
Bewertung der Deckung auf Kontoebene 529
Bewertung der Abdeckung von Amazon EC2 EC2-Instances 529
Statuswerte Amazon EC2 EC2-Instances 530
Bewertung der Abdeckung von Amazon ECR-Repositorien 533
Scanstatuswerte des Amazon ECR-Repositorys 534
Bewertung der Reichweite von Amazon ECR-Container-Images 535
Statuswerte für das Scannen von Amazon ECR-Container-Images 536
Bewertung des AWS Lambda Funktionsumfangs 537
Lambda-Funktionen scannen Statuswerte 538
Verwalten mehrerer Konten 539
Grundlegendes zum delegierten Administratorkonto und Mitgliedskonto 539
Modell zur Steuerung der Unternehmensrichtlinien 540
Delegierte Administratoraktionen 540
Aktionen für Mitgliedskonten 542
Benennen eines Administratorkontos 543
Überlegungen 543
Erforderliche Berechtigungen zum designieren eines delegierten Administrators 544
Benennen eines delegierten Administrators 545
Die Aktivierung von Amazon Inspector scannt nach Mitgliedskonten 547
Verknüpfung von Mitgliedskonten aufheben 551
Den delegierten Administrator entfernen 552
Taggen von -Ressourcen 554
Grundlagen des Kennzeichnens 554

Hinzufügen von -Tags ... 555
Hinzufügen von Tags zu Amazon Inspector Inspector-Ressourcen ... 555
Entfernen von Tags ... 557
Tags aus Amazon Inspector Inspector-Ressourcen entfernen ... 557
Usage ... 559
Verwenden der Nutzungskonsole ... 559
Erfahren Sie, wie Amazon Inspector die Nutzungskosten berechnet ... 562
Über die kostenlose Amazon Inspector-Testversion ... 563
Sicherheit ... 564
Datenschutz ... 565
Verschlüsselung im Ruhezustand ... 566
Verschlüsselung während der Übertragung ... 571
Identitäts- und Zugriffsverwaltung ... 571
Zielgruppe ... 572
Authentifizierung mit Identitäten ... 572
Verwalten des Zugriffs mit Richtlinien ... 574
So arbeitet Amazon Inspector mit IAM ... 576
Identity-based Beispiele für Richtlinien ... 581
AWS verwaltete Richtlinien ... 586
Verwenden von servicegebundenen Rollen ... 606
Fehlerbehebung ... 616
Überwachung von Amazon Inspector ... 618
CloudTrail protokolliert ... 619
Compliance-Validierung ... 622
Ausfallsicherheit ... 623
Sicherheit der Infrastruktur ... 623
Vorfallreaktion ... 624
Sicherheitsforschung ... 624
Forschungsmethodik ... 625
Zusammenfassung der Entdeckung ... 625
Aktuelle Berichte über bösartige Pakete ... 626
Zugehörige Ressourcen ... 627
AWS PrivateLink ... 627
Überlegungen ... 628
Erstellen eines Schnittstellenendpunkts ... 628
Integrationen ... 629

Verwenden von Amazon Inspector mit AWS Organizations ... 629
Integration von Amazon Inspector mit Amazon ECR ... 629
Integration von Amazon Inspector mit Security Hub CSPM ... 630
Amazon ECR-Integration ... 630
Aktivierung der Integration ... 630
Verwendung der Integration in einer Umgebung mit mehreren Konten ... 631
Integration in Security Hub CSPM ... 631
Ergebnisse von Amazon Inspector anzeigen in AWS Security Hub CSPM ... 632
Aktivierung und Konfiguration der Amazon Inspector Inspector-Integration mit Security Hub CSPM ... 635
Aktivierung von Amazon Inspector über Security Hub CSPM mithilfe der Unternehmensrichtlinie ... 636
Deaktivierung des Flusses von Ergebnissen aus einer Integration ... 636
Sicherheitskontrollen für Amazon Inspector in Security Hub CSPM anzeigen ... 636
Unterstützte Betriebssysteme und Programmiersprachen ... 638
Unterstützte Betriebssysteme ... 639
Unterstützte Betriebssysteme: Amazon EC2-Scanning ... 640
Unterstützte Betriebssysteme: Amazon ECR-Scannen mit Amazon Inspector ... 644
Unterstützte Betriebssysteme: CIS-Scanning ... 648
Unterstützte Betriebssysteme: Amazon Inspector Scan API ... 649
Eingestellte Betriebssysteme ... 651
Unterstützte Programmiersprachen ... 656
Unterstützte Programmiersprachen: Agentenloses Scannen mit Amazon EC2 ... 656
Unterstützte Programmiersprachen: Amazon Inspector VM Scanner ... 657
Unterstützte Programmiersprachen: Amazon Inspector SSM-Plugin ... 657
Unterstützte Programmiersprachen: Amazon ECR Scanning ... 658
Unterstützte Laufzeiten ... 658
Unterstützte Laufzeiten: Amazon Inspector Lambda Standard-Scanning ... 659
Unterstützte Laufzeiten: Amazon Inspector Lambda-Code-Scanning ... 660
Amazon Inspector deaktivieren ... 662
Deaktivierung von Amazon Inspector, der von Unternehmensrichtlinien verwaltet wird ... 663
Amazon Inspector deaktivieren ... 664
Kontingente ... 666
Regionen und Endpunkte ... 668
Service-Endpunkte für Amazon Inspector ... 668
Endpunkte für die Amazon Inspector Scan API ... 668

Region-specific Verfügbarkeit von Funktionen 677
Dokumentverlauf 681
Amazon Inspector Produktupdates 681
AWS Glossar 727
..... dccxxviii

Was ist Amazon Inspector?

Amazon Inspector ist ein Schwachstellen-Management-Service, der Workloads automatisch erkennt und sie kontinuierlich auf Softwareschwachstellen und unbeabsichtigte Netzwerkbedrohungen überprüft. Amazon Inspector erkennt und scannt Amazon EC2 EC2-Instances, Container-Images in Amazon ECR und Lambda-Funktionen. Wenn Amazon Inspector eine Sicherheitslücke in Software oder eine unbeabsichtigte Netzwerkgefährdung erkennt, erstellt Amazon Inspector einen Befund, bei dem es sich um einen detaillierten Bericht über das Problem handelt. Sie können Ergebnisse in der Amazon Inspector Inspector-Konsole oder API verwalten.

> **Note**
>
> Beim Einreichen einer Support-Anfrage kann Amazon Inspector auf relevante Ergebnisse an dem Ort, AWS-Region an dem sie gespeichert sind (aber innerhalb derselben Region), zugreifen und diese verarbeiten, um das Problem zu lösen.

Themen

- Funktionen von Amazon Inspector
- Zugreifen auf Amazon Inspector

Funktionen von Amazon Inspector

Zentrales Verwalten mehrerer Amazon Inspector Inspector-Konten

Wenn Ihre AWS Umgebung über mehrere Konten verfügt, können Sie Ihre Umgebung mithilfe von AWS Organizations über ein einziges Konto zentral verwalten. Mit diesem Ansatz können Sie ein Konto als delegiertes Administratorkonto für Amazon Inspector festlegen.

Amazon Inspector kann mit einem einzigen Klick für Ihr gesamtes Unternehmen aktiviert werden. Darüber hinaus können Sie die Aktivierung des Dienstes für future Mitglieder automatisieren, wann immer diese Ihrer Organisation beitreten. Das delegierte Administratorkonto von Amazon Inspector kann Ergebnisdaten und bestimmte Einstellungen für Mitglieder der Organisation verwalten. Dazu gehören die Anzeige aggregierter Ergebnisdetails für alle Mitgliedskonten, die Aktivierung oder Deaktivierung von Scans für Mitgliedskonten und die Überprüfung gescannter Ressourcen innerhalb der Organisation. AWS

Scannen Sie Ihre Umgebung kontinuierlich auf Sicherheitslücken und Netzwerkgefährdungen

Mit Amazon Inspector müssen Sie Bewertungsscans nicht manuell planen oder konfigurieren. Amazon Inspector erkennt automatisch Ihre in Frage kommenden Ressourcen und beginnt mit dem Scannen. Amazon Inspector bewertet Ihre Umgebung weiterhin während des gesamten Lebenszyklus Ihrer Ressourcen, indem es automatisch Ressourcen als Reaktion auf Änderungen, die zu einer neuen Sicherheitslücke führen könnten, erneut scannt, z. B.: Installation eines neuen Pakets in einer EC2-Instance, Installation eines Patches und wenn neue Common Vulnerabilities and Exposures (CVE), die sich auf die Ressource auswirken, veröffentlicht wird. Im Gegensatz zu herkömmlicher Sicherheitsscan-Software hat Amazon Inspector nur minimale Auswirkungen auf die Leistung Ihrer Flotte.

Wenn Sicherheitslücken oder offene Netzwerkpfade identifiziert werden, erstellt Amazon Inspector ein Ergebnis, das Sie untersuchen können. Das Ergebnis umfasst umfassende Informationen über die Sicherheitsanfälligkeit, die betroffene Ressource und Empfehlungen zur Behebung. Wenn Sie ein Ergebnis angemessen korrigieren, erkennt Amazon Inspector die Behebung automatisch und schließt das Ergebnis.

Beurteilen Sie Sicherheitslücken genau mit dem Amazon Inspector Risk Score

Amazon Inspector sammelt mithilfe von Scans Informationen über Ihre Umgebung und bietet Schweregrade, die speziell auf Ihre Umgebung zugeschnitten sind. Amazon Inspector untersucht die Sicherheitsmetriken, die den Basiswert der National Vulnerability Database (NVD) für eine Sicherheitslücke bilden, und passt sie an Ihre Computerumgebung an. Beispielsweise kann der Service den Amazon Inspector-Score eines Ergebnisses für eine Amazon EC2 EC2-Instance senken, wenn die Sicherheitsanfälligkeit über das Netzwerk ausgenutzt werden kann, aber von der Instance aus kein offener Netzwerkpfad zum Internet verfügbar ist. Diese Bewertung ist im CVSS-Format und ist eine Modifikation der von NVD bereitgestellten Basisbewertung des Common Vulnerability Scoring System (CVSS).

Identifizieren Sie wichtige Ergebnisse mit dem Amazon Inspector-Dashboard

Das Amazon Inspector-Dashboard bietet einen umfassenden Überblick über die Ergebnisse aus Ihrer gesamten Umgebung. Über das Dashboard können Sie auf die detaillierten Details eines Ergebnisses zugreifen. Das Dashboard enthält übersichtliche Informationen zur Scanabdeckung in Ihrer Umgebung, zu Ihren wichtigsten Ergebnissen und zu den Ressourcen, bei denen die meisten Ergebnisse vorliegen. Das Fenster zur risikobasierten Behebung im Amazon Inspector-Dashboard zeigt die Ergebnisse, die sich auf die größte Anzahl von Instances und Images auswirken.

Dieses Fenster erleichtert es, die Ergebnisse mit den größten Auswirkungen auf Ihre Umgebung zu identifizieren, die Einzelheiten der Ergebnisse zu überprüfen und Lösungsvorschläge zu überprüfen.

Verwalten Sie Ihre Ergebnisse mithilfe anpassbarer Ansichten

Zusätzlich zum Dashboard bietet die Amazon Inspector Inspector-Konsole eine Ergebnisansicht. Diese Seite listet alle Ergebnisse für Ihre Umgebung auf und enthält Einzelheiten zu den einzelnen Ergebnissen. Sie können die Ergebnisse nach Kategorie oder Schwachstellentyp gruppiert anzeigen. In jeder Ansicht können Sie Ihre Ergebnisse mithilfe von Filtern weiter anpassen. Sie können Filter auch verwenden, um Unterdrückungsregeln zu erstellen, die unerwünschte Ergebnisse in Ihren Ansichten verbergen.

Sie können Filter und Unterdrückungsregeln verwenden, um Ergebnisberichte zu erstellen, in denen alle Ergebnisse oder eine benutzerdefinierte Auswahl von Ergebnissen angezeigt werden. Berichte können im CSV- oder JSON-Format generiert werden.

Überwachen und verarbeiten Sie Ergebnisse mit anderen Diensten und Systemen

Um die Integration mit anderen Diensten und Systemen zu unterstützen, veröffentlicht Amazon Inspector die Ergebnisse in Form von Befundereignissen auf Amazon EventBridge. EventBridge ist ein serverloser Eventbus-Service, der Ergebnisdaten an Ziele wie AWS Lambda Funktionen und Amazon Simple Notification Service (Amazon SNS) -Themen weiterleiten kann. Damit EventBridge können Sie die Ergebnisse im Rahmen Ihrer bestehenden Sicherheits- und Compliance-Workflows nahezu in Echtzeit überwachen und verarbeiten.

Wenn Sie aktiviert haben AWS Security Hub CSPM, veröffentlicht Amazon Inspector die Ergebnisse auch im Security Hub CSPM. Security Hub CSPM ist ein Service, der einen umfassenden Überblick über Ihren Sicherheitsstatus in Ihrer gesamten AWS Umgebung bietet und Sie dabei unterstützt, Ihre Umgebung anhand von Branchenstandards und Best Practices zu überprüfen. Mit Security Hub CSPM können Sie Ihre Ergebnisse im Rahmen einer umfassenderen Analyse der Sicherheitslage Ihres Unternehmens in einfacher überwachen und verarbeiten. AWS

Zugreifen auf Amazon Inspector

Amazon Inspector ist in den meisten Fällen verfügbar AWS-Regionen. Eine Liste der Regionen, in denen Amazon Inspector derzeit verfügbar ist, finden Sie unter Amazon Inspector Inspector-Endpunkte und Kontingente in der Amazon Web Services General Reference. Weitere Informationen AWS-Regionen dazu finden Sie unter Managing AWS-Regionen in der Amazon Web Services General Reference. In jeder Region können Sie auf folgende Weise mit Amazon Inspector arbeiten.

AWS Management-Konsole

Die AWS-Managementkonsole ist eine browserbasierte Oberfläche, mit der Sie AWS Ressourcen erstellen und verwalten können. Als Teil dieser Konsole bietet die Amazon Inspector Inspector-Konsole Zugriff auf Ihr Amazon Inspector Inspector-Konto und Ihre Ressourcen. Sie können Amazon Inspector Inspector-Aufgaben von der Amazon Inspector Inspector-Konsole aus ausführen.

AWS Befehlszeilentools

Mit AWS Befehlszeilentools können Sie Befehle an der Befehlszeile Ihres Systems ausgeben, um Amazon Inspector Inspector-Aufgaben auszuführen. Die Verwendung der Befehlszeile kann schneller und bequemer sein als die Verwendung der Konsole. Die Befehlszeilen-Tools können auch beim Erstellen von Skripts für -Aufgaben hilfreich sein.

AWS stellt zwei Gruppen von Befehlszeilentools bereit: das AWS Command Line Interface (AWS CLI) und das AWS -Tools für PowerShell. Informationen zur Installation und Verwendung von finden Sie im AWS Command Line Interface User Guide. AWS CLI Informationen zur Installation und Verwendung der Tools für PowerShell finden Sie im AWS -Tools für PowerShell Benutzerhandbuch.

AWS SDKs

AWS bietet SDKs Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen, darunter Java, Go, Python, C++ und .NET. SDKs Sie bieten bequemen, programmatischen Zugriff auf Amazon Inspector und andere AWS-Services. Sie erledigen auch Aufgaben wie das kryptografische Signieren von Anfragen, das Verwalten von Fehlern und das automatische Wiederholen von Anfragen. Informationen zur Installation und Verwendung von finden Sie unter Tools AWS SDKs, auf denen Sie aufbauen können. AWS

Amazon Inspector REST-API

Die Amazon Inspector REST-API bietet Ihnen umfassenden, programmatischen Zugriff auf Ihr Amazon Inspector Inspector-Konto und Ihre Ressourcen. Mit dieser API können Sie HTTPS-Anfragen direkt an Amazon Inspector senden. Im Gegensatz zu den AWS Befehlszeilentools und SDKs bei der Verwendung dieser API muss Ihre Anwendung jedoch Details auf niedriger Ebene verarbeiten, z. B. die Generierung eines Hashs zum Signieren einer Anfrage.

Erste Schritte mit Amazon Inspector

Dieser Abschnitt enthält Informationen, die Sie vor der Aktivierung von Amazon Inspector berücksichtigen sollten, sowie ein Tutorial für die ersten Schritte, in dem beschrieben wird, wie Sie Amazon Inspector aktivieren und Ihre Ergebnisse in der Amazon Inspector Inspector-Konsole und mit der Amazon Inspector Inspector-API anzeigen können.

Themen

- Vor der Aktivierung von Amazon Inspector
- Tutorial „Erste Schritte“: Amazon Inspector aktivieren

Vor der Aktivierung von Amazon Inspector

Bevor Sie Amazon Inspector aktivieren, sollten Sie Folgendes beachten:

Amazon Inspector ist ein regionaler Service

Ihre Daten werden dort gespeichert, AWS-Region wo Sie Amazon Inspector aktivieren. Wiederholen Sie die Schritte im ersten Teil des Tutorials „Erste Schritte“ für alle, AWS-Regionen in denen Sie Amazon Inspector verwenden möchten.

Amazon Inspector erstellt die serviceverknüpften Rollen AWSService RoleForAmazonInspector 2 und 2Agentless AWSService RoleForAmazonInspector

Eine serviceverknüpfte Rolle ist eine Rolle in AWS Identity and Access Management (IAM), die mit einem Dienst verknüpft ist. AWS AWSServiceRoleForAmazonInspector2 und AWSServiceRoleForAmazonInspector2Agentless ermöglichen Amazon Inspector den Zugriff, der für die Durchführung von Sicherheitsbewertungen AWS-Services erforderlich ist.

IAM-Identitäten mit Administratorrechten können Amazon Inspector aktivieren

Schützen Sie Ihre Anmeldeinformationen, indem Sie Benutzer mit IAM oder erstellen. AWS IAM Identity Center Auf diese Weise können Sie sicherstellen, dass Benutzer nur über die Berechtigungen verfügen, die für die Verwaltung von Amazon Inspector erforderlich sind. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinie: AmazonInspectorFullAccess.

Hybrid-Scan wird automatisch aktiviert

Hybrid-Scanning umfasst agentenbasiertes Scannen und agentenloses Scannen. Standardmäßig verwendet Amazon Inspector diese Scanmethoden für alle in Frage kommenden EC2 Amazon-Instances. Weitere Informationen finden Sie unter EC2 Amazon-Instances mit Amazon Inspector scannen.

Für Amazon ECR- und Lambda-Funktionsscans ist kein SSM-Agent erforderlich

Beim agentenbasierten Scannen wird der SSM-Agent verwendet, um das Softwareinventar zu sammeln. Beim agentenlosen Scannen werden Amazon EBS-Snapshots verwendet, um Softwareinventar zu sammeln.

> **Note**
>
> Standardmäßig ist der SSM-Agent bereits in EC2 Amazon-Instances installiert, die auf Amazon Machine Images basieren. In einigen Fällen müssen Sie den SSM-Agenten jedoch möglicherweise manuell aktivieren. Weitere Informationen finden Sie im AWS Systems Manager Benutzerhandbuch unter Arbeiten mit dem SSM-Agenten.

Die monatlichen Kosten basieren auf den gescannten Workloads

Weitere Informationen erhalten Sie unter Amazon Inspector: Preise.

Aktivierung mehrerer Konten mit AWS Organizations

Für Organisationen, die Amazon Inspector verwenden AWS Organizations, unterstützt Amazon Inspector sowohl die delegierte Administratorverwaltung als auch die Aktivierung auf der Grundlage von Organisationsrichtlinien. Unternehmensrichtlinien bieten eine zentrale Verwaltung mit automatischer Aktivierung für neue Konten. Ausführliche Anweisungen zu beiden Ansätzen finden Sie unter. Tutorial „Erste Schritte“: Amazon Inspector aktivieren

Tutorial „Erste Schritte“: Amazon Inspector aktivieren

In diesem Thema wird beschrieben, wie Amazon Inspector für eine eigenständige Kontoumgebung (Mitgliedskonto) und eine Umgebung mit mehreren Konten (delegiertes Administratorkonto) aktiviert wird. Wenn Sie Amazon Inspector aktivieren, erkennt Amazon Inspector automatisch Workloads und scannt sie auf Softwareschwachstellen und unbeabsichtigte Netzwerkgefährdung.

Standalone account environment

Das folgende Verfahren beschreibt, wie Sie Amazon Inspector in der Konsole für ein Mitgliedskonto aktivieren. Um Amazon Inspector programmatisch zu aktivieren, inspector2 -. enablement-with-cli

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie Get Started.
3. Wählen Sie Amazon Inspector aktivieren.

Wenn Sie Amazon Inspector für ein eigenständiges Konto aktivieren, sind standardmäßig alle Scantypen aktiviert. Informationen zu Mitgliedskonten finden Sie unter Grundlegendes zum delegierten Administratorkonto und zu Mitgliedskonten in Amazon Inspector.

Multi-account (with AWS Organizations policy)

AWS Organizations Richtlinien bieten eine zentrale Steuerung, um Amazon Inspector in Ihrer gesamten Organisation zu aktivieren. Wenn Sie eine Unternehmensrichtlinie verwenden, wird die Aktivierung von Amazon Inspector automatisch für alle von der Richtlinie abgedeckten Konten verwaltet, und Mitgliedskonten können richtlinienverwaltete Scans mithilfe der Amazon Inspector Inspector-API nicht ändern.

Voraussetzungen

- Ihr Konto muss Teil einer Organisation sein. AWS Organizations
- Sie benötigen die erforderlichen Berechtigungen zum Erstellen und Verwalten von Organisationsrichtlinien in AWS Organizations.
- Der vertrauenswürdige Zugriff für Amazon Inspector muss in aktiviert sein AWS Organizations. Anweisungen finden Sie im AWS Organizations Benutzerhandbuch unter Enabling Trusted Access for Amazon Inspector.
- Die mit dem Amazon Inspector Service verknüpften Rollen sollten im Verwaltungskonto vorhanden sein. Um sie zu erstellen, aktivieren Sie Amazon Inspector im Verwaltungskonto oder führen Sie die folgenden Befehle vom Verwaltungskonto aus:
  - `aws iam create-service-linked-role --aws-service-name inspector2.amazonaws.com`
  - `aws iam create-service-linked-role --aws-service-name agentless.inspector2.amazonaws.com`

- Es sollte ein von Amazon Inspector delegierter Administrator benannt werden.

> **Note**
>
> Ohne die serviceverknüpften Amazon Inspector-Rollen Verwaltungskonto und delegierter Administrator wird durch die Unternehmensrichtlinien die Aktivierung von Amazon Inspector erzwungen, aber Mitgliedskonten werden nicht mit der Amazon Inspector Inspector-Organisation verknüpft, um zentrale Ergebnisse und Kontoverwaltung zu ermöglichen.

So aktivieren Sie Amazon Inspector mithilfe von AWS Organizations Richtlinien

1. Benennen Sie einen delegierten Administrator für Amazon Inspector, bevor Sie Organisationsrichtlinien erstellen, um sicherzustellen, dass Mitgliedskonten mit der Amazon Inspector Inspector-Organisation verknüpft sind, um die Ergebnisse zentral sichtbar zu machen. Melden Sie sich beim AWS Organizations Verwaltungskonto an, öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home und folgen Sie den Anweisungen unter. Benennen eines delegierten Administrators für Ihr AWS Organisation

   > **Note**
   >
   > Wir empfehlen dringend, Ihre AWS Organizations Amazon Inspector-Administrator-Konto-ID und die von Amazon Inspector designierte delegierte Administratorkonto-ID identisch zu lassen. Wenn sich die AWS Organizations delegierte Administratorkonto-ID von der delegierten Administratorkonto-ID von Amazon Inspector unterscheidet, priorisiert Amazon Inspector die von Inspector festgelegte Konto-ID. Wenn der delegierte Amazon Inspector-Administrator nicht eingerichtet ist, aber der AWS Organizations delegierte Administrator eingerichtet ist und das Verwaltungskonto die mit dem Amazon Inspector-Service verknüpften Rollen hat, weist Amazon Inspector automatisch die AWS Organizations delegierte Administratorkonto-ID als delegierten Amazon Inspector-Administrator zu.

2. Navigieren Sie in der Amazon Inspector Inspector-Konsole vom Verwaltungskonto aus zu Allgemeine Einstellungen. Wählen Sie unter Delegierungsrichtlinie die Option Kontoauszug anhängen aus. Überprüfen Sie im Dialogfeld Richtlinienerklärung anhängen die Richtlinie,

wählen Sie Ich bestätige, dass ich die Richtlinie überprüft habe und die damit gewährten Berechtigungen verstanden habe, und wählen Sie dann Erklärung anhängen aus.

> **Important**
>
> Das Verwaltungskonto muss über die folgenden Berechtigungen verfügen, um die Erklärung zur Delegierungsrichtlinie anhängen zu können:
>
> - Amazon Inspector Inspector-Berechtigungen aus der FullAccessverwalteten AmazonInspector 2-v2-Richtlinie
> - AWS Organizations `organizations:PutResourcePolicy`Erlaubnis aus der AWSOrganizationsFullAccessverwalteten Richtlinie
>
> Wenn die `organizations:PutResourcePolicy` Berechtigung fehlt, schlägt der Vorgang mit dem folgenden Fehler fehl:`Failed to attach statement to the delegation policy`.

3. Erstellen Sie als Nächstes eine Amazon Inspector AWS Organizations Inspector-Richtlinie. Wählen Sie im Navigationsbereich Management und dann Configurations aus.
4. Konfigurieren Sie die Richtlinie für das Schwachstellenmanagement. Geben Sie Details mit Namen und Beschreibung (optional) für die Richtlinie an.
5. Geben Sie auf der Seite Configure Inspector im Abschnitt Details einen Namen und eine Beschreibung für die Richtlinie ein. Führen Sie in der Funktionsauswahl einen der folgenden Schritte aus:

   - Wählen Sie „Konfigurieren“ und „Alle Funktionen aktivieren“ (empfohlen). Dadurch werden alle Inspector-Funktionen aktiviert, einschließlich EC2, ECR, Lambda-Standard, Lambda-Codescan und Code Security.
   - Wählen Sie „Teilmenge der Funktionen auswählen“. Wählen Sie alle Funktionen des Scantyps aus, die aktiviert werden sollen.
6. Wählen Sie im Bereich Kontoauswahl eine der folgenden Optionen aus:

   - Wählen Sie Alle Organisationseinheiten und Konten aus, wenn Sie die Konfiguration auf alle Organisationseinheiten und Konten anwenden möchten.
   - Wählen Sie Bestimmte Organisationseinheiten und Konten aus, wenn Sie die Konfiguration auf bestimmte Organisationseinheiten und Konten anwenden möchten. Wenn Sie diese

Option wählen, verwenden Sie die Suchleiste oder den Organisationsstrukturbaum, um die Organisationseinheiten und Konten anzugeben, auf die die Richtlinie angewendet werden soll.

- Wählen Sie Keine Organisationseinheiten oder Konten aus, wenn Sie die Konfiguration nicht auf eine Organisationseinheit oder ein Konto anwenden möchten.

7. Wählen Sie im Abschnitt Regionen die Optionen Alle Regionen aktivieren, Alle Regionen deaktivieren oder Regionen angeben aus.

   - Wenn Sie „Alle Regionen aktivieren“ wählen, können Sie festlegen, ob neue Regionen automatisch aktiviert werden sollen.
   - Wenn Sie Alle Regionen deaktivieren wählen, können Sie festlegen, ob neue Regionen automatisch deaktiviert werden sollen.
   - Wenn Sie „Regionen angeben“ wählen, müssen Sie auswählen, welche Regionen Sie aktivieren und deaktivieren möchten.

   (Optional) Informationen zu den erweiterten Einstellungen finden Sie in der Anleitung von AWS Organizations.

   (Optional) Fügen Sie für Resource-Tags Tags als Schlüssel-Wert-Paare hinzu, damit Sie die Konfiguration leichter identifizieren können.

8. Wählen Sie Weiter, überprüfen Sie Ihre Änderungen und wählen Sie dann Anwenden aus. Ihre Zielkonten werden auf der Grundlage der Richtlinie konfiguriert. Der Konfigurationsstatus Ihrer Richtlinie wird oben auf der Seite Richtlinien angezeigt. Jede Funktion gibt einen Status an, der angibt, ob sie konfiguriert wurde oder wo Bereitstellungsfehler aufgetreten sind. Wählen Sie bei Fehlern den Link für die Fehlermeldung aus, um weitere Informationen zu erhalten. Die geltenden Richtlinien auf Kontoebene finden Sie auf der Seite Konfigurationen auf der Registerkarte Organisation, auf der Sie ein Konto auswählen können.

Wenn Amazon Inspector durch Unternehmensrichtlinien aktiviert ist, können Konten, die unter die Richtlinie fallen, die richtlinienverwalteten Scantypen nicht über die Amazon Inspector Inspector-API oder -Konsole deaktivieren. Ausführliche Informationen darüber, was delegierte Administratoren und Mitgliedskonten gemäß den Unternehmensrichtlinien tun können und was nicht, finden Sie unter. Verwaltung mehrerer Konten in Amazon Inspector mit AWS Organizations

Multi-account (without AWS Organizations policy)

> **Note**
>
> Sie müssen das AWS Organizations Verwaltungskonto verwenden, um dieses Verfahren abzuschließen. Nur das AWS Organizations Verwaltungskonto kann einen delegierten Administrator benennen. Für die Benennung eines delegierten Administrators sind möglicherweise Berechtigungen erforderlich. Weitere Informationen finden Sie unter Erforderliche Berechtigungen zum designieren eines delegierten Administrators.

Wenn Sie Amazon Inspector zum ersten Mal aktivieren, erstellt Amazon Inspector die serviceverknüpfte Rolle `AWSServiceRoleForAmazonInspector` für das Konto. Informationen darüber, wie Amazon Inspector serviceverknüpfte Rollen verwendet, finden Sie unterVerwenden von serviceverknüpften Rollen für Amazon Inspector.

So benennen Sie einen delegierten Administrator für Amazon Inspector

1. Melden Sie sich beim AWS Organizations Verwaltungskonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie Erste Schritte.
3. Geben Sie unter Delegierter Administrator die 12-stellige ID des Benutzers ein, den AWS-Konto Sie als delegierten Administrator festlegen möchten.
4. Wählen Sie Delegieren und dann erneut Delegieren aus.
5. (Optional) Wenn Sie Amazon Inspector für das AWS Organizations Verwaltungskonto aktivieren möchten, wählen Sie unter Serviceberechtigungen die Option Amazon Inspector aktivieren aus.

Wenn Sie einen delegierten Administrator benennen, sind standardmäßig alle Scantypen für das Konto aktiviert. Informationen zum delegierten Administratorkonto finden Sie unter Grundlegendes zum delegierten Administratorkonto und zu Mitgliedskonten in Amazon Inspector.

Automatisierte Scantypen in Amazon Inspector

Amazon Inspector verwendet eine speziell entwickelte Scan-Engine, die Ihre Ressourcen auf umsetzbare Softwareschwachstellen und unbeabsichtigte Netzwerkbedrohungen überwacht. Wenn Amazon Inspector eine Sicherheitslücke in Software oder eine unbeabsichtigte Netzwerkgefährdung erkennt, wird ein Ergebnis erstellt. Wenn Sie Amazon Inspector zum ersten Mal aktivieren, wird Ihr Konto automatisch für alle Scantypen registriert, darunter Amazon Amazon EC2-Scannen, Amazon ECR-Scanning und Lambda-Standardscans.

> **Note**
>
> Das Lambda-Code-Scannen ist eine optionale Ebene des Lambda-Funktionsscannens, die Sie jederzeit aktivieren können.

Themen

- Übersicht der Amazon Inspector-Scantypen
- Einen Scantyp aktivieren
- Scannen von Amazon EC2 EC2-Instances mit Amazon Inspector
- Scannen von Amazon Elastic Container Registry-Container-Images mit Amazon Inspector
- Scanning AWS Lambda funktioniert mit Amazon Inspector
- Deaktivieren eines Scantyps in Amazon Inspector

Übersicht der Amazon Inspector-Scantypen

Amazon Inspector bietet verschiedene Scantypen, die sich auf bestimmte Ressourcentypen in Ihrer AWS Umgebung konzentrieren.

Amazon EC2-Scannen

Wenn Sie Amazon EC2-Scanning aktivieren, scannt Amazon Inspector Ihre EC2-Instances auf allgemeine Sicherheitslücken und Exposures (CVEs), Netzwerkprobleme, Probleme mit der Netzwerkerreichbarkeit sowie Sicherheitslücken im Betriebssystem und in Programmiersprachenpaketen. Amazon Inspector führt Scans mithilfe des auf Ihrer Instance installierten SSM-Agenten oder mithilfe von Amazon EBS-Snapshots von Instances durch. Weitere Informationen finden Sie unter Scannen von Amazon EC2 EC2-Instances mit Amazon

Inspector. Wenn Sie Amazon EC2-Scannen aktivieren, aktivieren Sie standardmäßig automatisch den Hybrid-Scanmodus. Weitere Informationen finden Sie unter Agentloses Scannen.

Amazon ECR-Scannen

Wenn Sie das Amazon ECR-Scannen aktivieren, konvertiert Amazon Inspector alle Repositorys in Ihrer privaten Registrierung von einfachen Scan-Container-Repositorys in erweiterte Scan-Repositorys. Sie können diese Einstellung mit Einschlussregeln so konfigurieren, dass nur On-Push-Scans oder ausgewählte Repositorys gescannt werden. Amazon Inspector scannt nur ECR-Container-Images, die in ECR aktiv sind (`imageStatus`Feld ist`ACTIVE`). Amazon Inspector scannt innerhalb der konfigurierten Dauer des erneuten Scans alle Bilder, die in ECR übertragen oder auf aktiv (`lastActivatedAt`) umgestellt oder abgerufen wurden. Die Standarddauer für das erneute Scannen neuer Konten, einschließlich neuer Konten, die zu einer Organisation hinzugefügt wurden, beträgt 14 Tage. Für Amazon Inspector Inspector-Konten, die vor dem 16. Mai 2024 erstellt wurden, überwacht die Standardkonfiguration Bilder, die innerhalb der letzten 90 Tage per Push oder Pull übertragen wurden. Sie können diese Einstellung jederzeit ändern. Weitere Informationen finden Sie unter Scannen von Amazon Elastic Container Registry-Container-Images mit Amazon Inspector.

Lambda-Standardscannen

Wenn Sie das Lambda-Standardscannen aktivieren, erkennt Amazon Inspector alle Lambda-Funktionen in Ihrem Konto und scannt sie sofort auf Sicherheitslücken. Amazon Inspector scannt neue Lambda-Funktionen und -Layer, wenn sie bereitgestellt werden. Amazon Inspector scannt sie erneut, wenn sie aktualisiert werden oder wenn neue CVEs veröffentlicht werden. Weitere Informationen zum Scannen finden Sie unter. Scanning AWS Lambda funktioniert mit Amazon Inspector

Lambda-Standardscannen + Lambda-Code-Scannen

Wenn Sie das Lambda-Code-Scannen aktivieren, erkennt Amazon Inspector die Lambda-Funktionen und -Layer in Ihrem Konto und scannt sie auf Code-Schwachstellen. Diese Art des Scannens bewertet Abhängigkeiten von Anwendungspaketen, die in einer Lambda-Funktion für CVEs verwendet werden. Wenn Sie diesen Scantyp aktivieren, aktivieren Sie auch den Lambda-Standardscan. Weitere Informationen finden Sie unter Scanning AWS Lambda funktioniert mit Amazon Inspector.

Codesicherheit für Amazon Inspector

Dieser Scantyp nutzt die Amazon Q Developer-Scan-Engine, um Anwendungscode von Erstanbietern, Abhängigkeiten von Drittanbieteranwendungen und Infrastructure as Code auf

Sicherheitslücken zu scannen. Weitere Informationen finden Sie unter Code Security for Amazon Inspector.

Einen Scantyp aktivieren

Sie können einen Scantyp jederzeit aktivieren. Wenn Sie einen Scan-Typ aktivieren, beginnt Amazon Inspector mit dem Scannen geeigneter Ressourcen für diesen Scantyp.

Amazon EC2-Scannen

Dieser Scantyp extrahiert Metadaten aus einer Amazon EC2 EC2-Instance, bevor die Metadaten mit Regeln verglichen werden, die in Sicherheitsempfehlungen gesammelt wurden. Wenn Sie diesen Scantyp aktivieren, scannt Amazon Inspector alle geeigneten Amazon EC2 EC2-Instances in Ihrem Konto auf Paketschwachstellen und Probleme mit der Netzwerkerreichbarkeit. Nachdem Sie diesen Scan-Typ aktiviert haben, können Sie auf der Registerkarte Instances sehen, wie viele Instances gescannt werden.

Amazon ECR-Scannen

Dieser Scantyp scannt Container-Images und Container-Repositorys in Amazon ECR. Wenn Sie diesen Scantyp aktivieren, ändern Sie die Einstellung der Scan-Konfiguration für Ihre private Registrierung von einfachem Scannen zu erweitertem Scannen. Nachdem Sie das Amazon ECR-Scannen aktiviert haben, können Sie auf den Registerkarten Container-Images und Container-Repositorys sehen, wie viele Bilder und Repositorys gescannt werden.

Lambda-Standardscannen + Scannen von Lambda-Code

Lambda-Standardscan ist der standardmäßige Lambda-Scantyp. Wenn Sie das Lambda-Standardscannen aktivieren, werden alle Ihre Lambda-Funktionen auf Softwareschwachstellen gescannt, sofern sie in den letzten 90 Tagen aufgerufen oder aktualisiert wurden. Nachdem Sie das Lambda-Standardscannen aktiviert haben, sehen Sie auf der Registerkarte Lambda-Funktionen, wie viele Lambda-Funktionen gescannt werden.

Das Scannen von Lambda-Code scannt benutzerdefinierten Anwendungscode in einer Lambda-Funktion. Wenn Sie das Lambda-Code-Scanning aktivieren, werden alle Ihre Lambda-Funktionen auf Code-Schwachstellen gescannt, sofern sie in den letzten 90 Tagen aufgerufen oder aktualisiert wurden. Nachdem Sie den Lambda-Standardscan aktiviert haben, können Sie auf der Registerkarte Lambda-Funktionen sehen, wie viele Lambda-Funktionen auf Code-Schwachstellen gescannt werden.

> **Note**
>
> Wenn Sie das Lambda-Code-Scannen aktivieren möchten, müssen Sie zuerst das Lambda-Standardscannen aktivieren.

[Amazon Inspector Codesicherheit]

Dieser Scantyp durchsucht Anwendungscode von Erstanbietern, Abhängigkeiten von Drittanbieteranwendungen und Infrastructure as Code auf Sicherheitslücken. Wenn Sie Code Security aktivieren, beginnt Amazon Inspector, Ihre Code-Repositorys auf der Grundlage Ihrer Scan-Konfigurationen nach Code-Schwachstellen zu durchsuchen. Nachdem Sie Amazon Inspector Code Security aktiviert haben, können Sie auf der Registerkarte Code-Repositorys sehen, wie viele Code-Repositorys gescannt werden.

Scans aktivieren

Das folgende Verfahren beschreibt, wie Sie einen Scantyp in Amazon Inspector aktivieren.

> **Note**
>
> Wenn Sie der delegierte Administrator für eine AWS Organisation sind, können Sie Amazon Inspector-Scantypen für mehrere Konten in mehreren Regionen mithilfe eines Shell-Skripts aktivieren. Weitere Informationen finden Sie unter inspector2-enablement-with-cli on. GitHub Andernfalls führen Sie die folgenden Schritte aus, während Sie als delegierter Amazon Inspector-Administrator angemeldet sind.

Console

Um Scans zu aktivieren

1. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mit AWS-Region der Auswahl in der oberen rechten Ecke der Seite die Region aus, in der Sie einen neuen Scantyp aktivieren möchten.
3. Wählen Sie im Navigationsbereich Kontoverwaltung aus.

4. Wählen Sie auf der Seite Kontoverwaltung die Konten aus, für die Sie einen Scantyp aktivieren möchten.
5. Wählen Sie Aktivieren und wählen Sie die Art des Scannens aus, den Sie aktivieren möchten.
6. (Empfohlen) Wiederholen Sie diese Schritte AWS-Region für jeden, für den Sie diesen Scantyp aktivieren möchten.

API

Führen Sie den Vorgang „API aktivieren“ aus. Geben Sie in der Anfrage die Konto-IDs an, für die Sie Scans aktivieren, und das Idempotenz-Token sowie eine oder mehrere von`EC2`,, oder `LAMBDA_CODE` für`ECR`, oder`LAMBDA`, `resourceTypes` um Scans dieses Typs zu aktivieren.

Scannen von Amazon EC2 EC2-Instances mit Amazon Inspector

Amazon Inspector Amazon EC2-Scanning extrahiert Metadaten aus Ihrer EC2-Instance, bevor die Metadaten mit Regeln verglichen werden, die in Sicherheitsempfehlungen gesammelt wurden. Amazon Inspector scannt Instances auf Sicherheitslücken in Paketen und Problemen mit der Erreichbarkeit des Netzwerks, um Ergebnisse zu erzielen. Amazon Inspector führt alle 12 Stunden Netzwerkerreichbarkeitsscans und Paketschwachstellenscans in einem variablen Rhythmus durch, der von der Scanmethode abhängt, die mit der EC2-Instance verknüpft ist.

Scans nach Sicherheitslücken in Paketen können mit einer agentenbasierten oder agentenlosen Scanmethode durchgeführt werden. Beide Scanmethoden bestimmen, wie und wann Amazon Inspector das Softwareinventar von einer EC2-Instance für Paketschwachstellenscans erfasst. Agent-based Beim Scannen wird Softwareinventar von Ihren Instances mithilfe des Amazon EC2 Systems Manager (SSM) -Agenten erfasst, und beim agentenlosen Scannen wird Softwareinventar mithilfe von Amazon EBS-Snapshots erfasst.

Amazon Inspector verwendet die Scanmethoden, die Sie für Ihr Konto aktivieren. Wenn Sie Amazon Inspector zum ersten Mal aktivieren, wird Ihr Konto automatisch für das Hybrid-Scannen registriert, bei dem beide Scanmethoden verwendet werden. Sie können diese Einstellung jedoch jederzeit ändern. Informationen zur Aktivierung eines Suchtyps finden Sie unter Einen Scantyp aktivieren. Dieser Abschnitt enthält Informationen zum Amazon EC2-Scannen.

Agent-based Scannen

Verbessertes EC2-Scannen

Verbessertes EC2-Scannen wird mit dem durchgeführt. Amazon Inspector VM-Scanner Dieser Scanner wird mithilfe von SSM-Verknüpfungen installiert und aktualisiert. Kunden können sich anmelden, indem sie zu ihrer Amazon Inspector Inspector-Konsole gehen und die Seite Einstellungen > Scaneinstellungen aufrufen. Wählen Sie Upgrade starten, um mit dem erweiterten EC2-Scannen zu beginnen.

> **Empfehlung**
>
> Wir empfehlen, auf Enhanced EC2 Scanning zu aktualisieren. Der Amazon Inspector VM Scanner verwendet denselben Scanmechanismus für alle Betriebssysteme und für andere von Amazon Inspector unterstützte Ressourcen, wodurch konsistentere Ergebnisse erzielt werden als das Amazon Inspector SSM-Plugin. WindowsInsbesondere bei werden Timeouts pro Abfrage vermieden, die dazu führen können, dass das Amazon Inspector SSM-Plugin Ergebnisse inkonsistent meldet.

1. Amazon Inspector erstellt SSM-Verknüpfungen in Ihrem Konto, um Inventar aus Ihren Instances zu sammeln. Diese Verknüpfungen installieren Plugins auf einzelnen Instances, um Inventar zu sammeln.
2. Mithilfe von Systemtools wie Systemd und Scheduled Tasks extrahiert Inspector VM Scanner das Paketinventar aus einer Instance und übermittelt diese Informationen an Amazon Inspector.
3. Amazon Inspector bewertet das extrahierte Inventar und generiert Ergebnisse für alle erkannten Sicherheitslücken.

Standard-Scannen

Agent-based Scans werden kontinuierlich mit dem Amazon Inspector SSM-Plugin auf allen geeigneten Instances durchgeführt. Für agentenbasierte Scans verwendet Amazon Inspector SSM-Verknüpfungen und über diese Verknüpfungen installierte Plugins, um Softwarebestand aus Ihren Instances zu sammeln. Zusätzlich zu Paket-Schwachstellenscans für Betriebssystempakete kann das agentenbasierte Scannen von Amazon Inspector auch Paketschwachstellen in Paketen in Programmiersprachenpaketen durch Amazon Inspector Deep Inspection erkennen.

Der folgende Prozess erklärt, wie Amazon Inspector SSM verwendet, um Inventar zu sammeln und agentenbasierte Scans durchzuführen:

1. Amazon Inspector erstellt SSM-Verknüpfungen in Ihrem Konto, um Inventar aus Ihren Instances zu sammeln. Diese Verknüpfungen installieren Plugins auf einzelnen Instances, um Inventar zu sammeln.
2. Mithilfe von SSM extrahiert Amazon Inspector Paketinventar aus einer Instance.
3. Amazon Inspector bewertet das extrahierte Inventar und generiert Ergebnisse für alle erkannten Sicherheitslücken.

> **Note**
>
> Für agentenbasiertes Scannen muss die Amazon EC2 EC2-Instance von SSM in derselben verwaltet werden. AWS-Konto

Amazon VPC-Endpunktanforderungen für erweitertes EC2-Scannen auf privaten Amazon EC2 EC2-Instances

Sie können Enhanced EC2 Scanning auf Amazon EC2 EC2-Instances über ein Amazon-Netzwerk ausführen. Wenn Sie jedoch Enhanced EC2 Scanning auf privaten Amazon EC2 EC2-Instances ausführen möchten, müssen Sie Amazon VPC-Endpoints erstellen. Die folgenden Endpunkte sind erforderlich:

- `com.amazonaws.region.inspector2-telemetry`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ssm`
- `com.amazonaws.region.ssmmessages`

Wo *`region`* ist der Regionalcode für den AWS-Region zutreffenden.

Weitere Informationen finden Sie unter Verbessern der Sicherheit von Amazon EC2 EC2-Instances mithilfe von Amazon VPC-Endpunkten für Systems Manager im AWS Systems Manager Benutzerhandbuch.

> **Note**
>
> Ab SSM Agent Version 3.3.40.0 verwendet SSM Agent den `ssmmessages` Endpunkt anstelle des Endpunkts, wann immer dieser verfügbar ist. `ec2messages` Frühere Versionen von SSM Agent verwenden den Endpunkt. `ec2messages` Dieser Endpunkt wird nur unterstützt, wenn er vor 2024 AWS-Regionen gestartet wurde. Weitere Informationen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen im Benutzerhandbuch.AWS Systems Manager

> **Note**
>
> Derzeit unterstützen einige den AWS-Regionen Endpunkt nicht. `com.amazonaws.region.inspector2-telemetry`

In Frage kommende Instanzen

Amazon Inspector verwendet die agentenbasierte Methode, um eine Instance zu scannen, wenn sie die folgenden Bedingungen erfüllt:

- Die Instance hat ein unterstütztes Betriebssystem. Eine Liste der unterstützten Betriebssysteme finden Sie in der Spalte Agent-based Scanunterstützung vonthe section called “Unterstützte Betriebssysteme: Amazon EC2-Scanning”.
- Die Instance wird nicht von Scans durch Amazon Inspector EC2-Ausschluss-Tags ausgeschlossen.

Agent-based Verhalten beim Scannen

Bei Verwendung der agentenbasierten Scanmethode initiiert Amazon Inspector in den folgenden Situationen neue Schwachstellenscans von EC2-Instances:

- Wenn Sie eine neue EC2-Instance starten.
- Wenn Sie neue Software auf einer vorhandenen EC2-Instance (Linux und Mac) installieren.
- Wenn Amazon Inspector seiner Datenbank ein neues CVE-Element (Common Vulnerabilities and Exposures) hinzufügt und dieses CVE für Ihre EC2-Instance (Linux und Mac) relevant ist.

Amazon Inspector aktualisiert das Feld Zuletzt gescannt für eine EC2-Instance, wenn ein erster Scan abgeschlossen ist. Danach wird das Feld Zuletzt gescannt aktualisiert, wenn Amazon Inspector das SSM-Inventar auswertet (standardmäßig alle 30 Minuten) oder wenn eine Instance erneut gescannt wird, weil ein neuer CVE, der sich auf diese Instance auswirkt, zur Amazon Inspector Inspector-Datenbank hinzugefügt wurde.

Sie können auf der Kontoverwaltungsseite auf der Registerkarte Instances oder mithilfe des Befehls überprüfen, wann eine EC2-Instance zuletzt auf Sicherheitslücken gescannt wurde. ListCoverage

Den SSM-Agenten konfigurieren

Damit Amazon Inspector mithilfe der agentenbasierten Scanmethode Softwareschwachstellen für eine Amazon EC2-Instance erkennen kann, muss es sich bei der Instance um eine verwaltete Instance in Amazon EC2 Systems Manager (SSM) handeln. Bei einer von SSM verwalteten Instance ist der SSM-Agent installiert und läuft, und SSM ist berechtigt, die Instance zu verwalten. Wenn Sie SSM bereits zur Verwaltung Ihrer Instanzen verwenden, sind für agentenbasierte Scans keine weiteren Schritte erforderlich.

Der SSM-Agent wird standardmäßig auf EC2-Instances installiert, die aus einigen Amazon Machine Images (AMIs) erstellt wurden. Weitere Informationen finden Sie unter Über SSM Agent im AWS Systems Manager Benutzerhandbuch. Selbst wenn er installiert ist, müssen Sie den SSM-Agenten möglicherweise manuell aktivieren und SSM die Berechtigung zur Verwaltung Ihrer Instanz erteilen.

Das folgende Verfahren beschreibt, wie Sie eine Amazon EC2 EC2-Instance mithilfe eines IAM-Instance-Profils als verwaltete Instance konfigurieren. Das Verfahren enthält auch Links zu detaillierteren Informationen im AWS Systems Manager Benutzerhandbuch.

AmazonSSMManagedInstanceCoreist die empfohlene Richtlinie, die Sie verwenden sollten, wenn Sie ein Instanzprofil anhängen. Diese Richtlinie verfügt über alle Berechtigungen, die für das Scannen mit Amazon Inspector EC2 erforderlich sind.

> **Note**
>
> Sie können die SSM-Verwaltung all Ihrer EC2-Instances auch ohne die Verwendung von IAM-Instance-Profilen automatisieren, indem Sie die SSM-Standardkonfiguration für die Hostverwaltung verwenden. Weitere Informationen finden Sie unter Standardkonfiguration für die Host-Verwaltung.

So konfigurieren Sie SSM für eine Amazon EC2 EC2-Instance

1. Wenn es noch nicht von Ihrem Betriebssystemanbieter installiert wurde, installieren Sie den SSM-Agent. Weitere Informationen finden Sie unter Arbeiten mit dem SSM-Agenten.
2. Verwenden Sie den AWS CLI , um zu überprüfen, ob der SSM-Agent ausgeführt wird. Weitere Informationen finden Sie unter Prüfen des Status des SSM-Agents und Starten des Agenten.
3. Erteilen Sie SSM die Erlaubnis, Ihre Instanz zu verwalten. Sie können die Erlaubnis erteilen, indem Sie ein IAM-Instanzprofil erstellen und es an Ihre Instanz anhängen. Wir empfehlen die Verwendung dieser AmazonSSMManagedInstanceCoreRichtlinie, da diese Richtlinie über die Berechtigungen für SSM Distributor, SSM Inventory und SSM State Manager verfügt, die Amazon Inspector für Scans benötigt. Anweisungen zum Erstellen eines Instanzprofils mit diesen Berechtigungen und zum Anhängen an eine Instanz finden Sie unter Instanzberechtigungen für Systems Manager konfigurieren.
4. (Optional) Aktivieren Sie automatische Updates für den SSM-Agent. Weitere Informationen finden Sie unter Automatisieren von Updates für den SSM-Agenten.
5. (Optional) Konfigurieren Sie Systems Manager für die Verwendung eines Amazon Virtual Private Cloud (Amazon VPC) -Endpunkts. Weitere Informationen finden Sie unter Amazon VPC-Endpoints erstellen.

Für das Scannen erstellte SSM-Ressourcen

Amazon Inspector benötigt eine Reihe von SSM-Ressourcen in Ihrem Konto, um Amazon EC2-Scans auszuführen. Die folgenden Ressourcen werden erstellt, wenn Sie das Amazon Inspector EC2-Scannen zum ersten Mal aktivieren:

> **Note**
>
> Wenn eine dieser SSM-Ressourcen gelöscht wird, während Amazon Inspector Amazon EC2-Scannen für Ihr Konto aktiviert ist, versucht Amazon Inspector, sie beim nächsten Scanintervall neu zu erstellen.

`InspectorInventoryCollection-do-not-delete`

Dies ist eine Systems Manager State Manager (SSM) -Zuordnung, die Amazon Inspector verwendet, um Softwareanwendungsinventar aus Ihren Amazon EC2 EC2-Instances zu

sammeln. Wenn Ihr Konto bereits über eine SSM-Verknüpfung für die Erfassung von Inventar verfügt`InstanceIds*`, verwendet Amazon Inspector diese, anstatt eine eigene zu erstellen.

`InspectorResourceDataSync-do-not-delete`

Dies ist eine Ressourcendatensynchronisierung, die Amazon Inspector verwendet, um gesammelte Inventardaten von Ihren Amazon EC2 EC2-Instances an einen Amazon S3 S3-Bucket zu senden, der Amazon Inspector gehört. Weitere Informationen finden Sie unter Konfiguration der Ressourcendatensynchronisierung für Inventar im AWS Systems Manager Benutzerhandbuch.

`InspectorVmScannerDistributor-do-not-delete`

Dies ist eine SSM-Verknüpfung, die Amazon Inspector verwendet, um die Amazon Inspector VM-Scanner auf Ihren Amazon EC2 EC2-Instances zu installieren und zu aktualisieren.

`InspectorDistributor-do-not-delete`

Dies ist eine SSM-Verknüpfung, die Amazon Inspector zum Scannen von Windows-Instances verwendet. Diese Assoziation installiert das Amazon Inspector SSM-Plugin auf Ihren Windows-Instances. Wenn die Plugin-Datei versehentlich gelöscht wird, wird sie durch diese Verknüpfung beim nächsten Zuordnungsintervall erneut installiert.

`InvokeInspectorSsmPlugin-do-not-delete`

Dies ist eine SSM-Verknüpfung, die Amazon Inspector zum Scannen von Windows-Instances verwendet. Diese Zuordnung ermöglicht es Amazon Inspector, Scans mithilfe des Plug-ins zu initiieren. Sie können damit auch benutzerdefinierte Intervalle für Scans von Windows-Instances festlegen. Weitere Informationen finden Sie unter Einstellung benutzerdefinierter Zeitpläne für Windows Instanzscans.

`InspectorLinuxDistributor-do-not-delete`

Dies ist eine SSM-Assoziation, die Amazon Inspector für Amazon EC2 Linux Deep Inspection verwendet. Diese Assoziation installiert das Amazon Inspector SSM-Plugin auf Ihren Linux-Instances.

`InvokeInspectorLinuxSsmPlugin-do-not-delete`

Dies ist eine SSM-Verbindung, die Amazon Inspector für Amazon EC2 Linux Deep Inspection verwendet. Diese Zuordnung ermöglicht es Amazon Inspector, Scans mithilfe des Plug-ins zu initiieren.

> **Note**
>
> Wenn Sie Amazon Inspector Amazon EC2 Scanning oder Deep Inspection deaktivieren, `InvokeInspectorLinuxSsmPlugin-do-not-delete` wird die SSM-Ressource nicht mehr aufgerufen.

Scannen ohne Agenten

Amazon Inspector verwendet die agentenlose Scanmethode für berechtigte Instances, wenn sich Ihr Konto im Hybrid-Scanmodus befindet. Der Hybrid-Scanmodus umfasst agentenbasierte und agentenlose Scans und wird automatisch aktiviert, wenn Sie das Amazon EC2-Scannen aktivieren.

Für Scans ohne Agenten verwendet Amazon Inspector EBS-Snapshots, um ein Softwareinventar aus Ihren Instances zu erfassen. Beim agentenlosen Scannen werden Instances nach Sicherheitslücken im Betriebssystem und in Paketen der Anwendungsprogrammiersprache durchsucht.

> **Note**
>
> Beim Scannen von Linux-Instances nach Sicherheitslücken in Paketen in der Programmiersprache werden mit der agentenlosen Methode alle verfügbaren Pfade gescannt, wohingegen das agentenbasierte Scannen nur die Standardpfade und zusätzliche Pfade scannt, die Sie als Teil angeben. Eingehende Inspektion von Amazon Inspector für Linux-based Amazon EC2 EC2-Instances Dies kann dazu führen, dass dieselbe Instanz unterschiedliche Ergebnisse erzielt, je nachdem, ob sie mit der agentenbasierten Methode oder der agentenlosen Methode gescannt wird.

Der folgende Prozess erklärt, wie Amazon Inspector EBS-Snapshots verwendet, um Inventar zu sammeln und agentenlose Scans durchzuführen:

1. Amazon Inspector erstellt einen EBS-Snapshot aller Volumes, die an die Instance angehängt sind. Während Amazon Inspector es verwendet, wird der Snapshot in Ihrem Konto gespeichert und mit `InspectorScan` einem Tag-Schlüssel und einer eindeutigen Scan-ID als Tag-Wert gekennzeichnet.
2. Amazon Inspector ruft mithilfe von EBS Direct-APIs Daten aus den Snapshots ab und bewertet sie auf Sicherheitslücken. Die Ergebnisse werden für alle erkannten Sicherheitslücken generiert.
3. Amazon Inspector löscht die EBS-Snapshots, die es in Ihrem Konto erstellt hat.

Qualifizierte Instances

Amazon Inspector verwendet die agentenlose Methode, um eine Instance zu scannen, wenn sie die folgenden Bedingungen erfüllt:

- Die Instance hat ein unterstütztes Betriebssystem. Weitere Informationen finden Sie in der Spalte > Agent-based Scanunterstützung vonthe section called “Unterstützte Betriebssysteme: Amazon EC2-Scanning”.
- Die Instanz hat den Status `Unmanaged EC2 instance`Stale `inventory`, oder`No inventory`.
- Die Instance wird von Amazon EBS unterstützt und hat eines der folgenden Dateisystemformate:
  - `ext3`
  - `ext4`
  - `xfs`
- Die Instance ist nicht von Scans über Amazon EC2 EC2-Ausschluss-Tags ausgeschlossen.
- Die Anzahl der an die Instance angehängten Volumes beträgt weniger als 8 und ihre Gesamtgröße beträgt höchstens 1200 GB.

Verhalten beim Scannen ohne Agenten

Wenn Ihr Konto für Hybrid-Scanning konfiguriert ist, führt Amazon Inspector alle 24 Stunden agentenlose Scans auf geeigneten Instances durch. Amazon Inspector erkennt und scannt jede Stunde neue infrage kommende Instances, einschließlich neuer Instances ohne SSM-Agenten oder bereits existierende Instances mit Status, der auf geändert wurde. `SSM_UNMANAGED`

Amazon Inspector aktualisiert das Feld Zuletzt gescannt für eine Amazon EC2 EC2-Instance, wenn nach einem agentenlosen Scan extrahierte Snapshots aus einer Instance gescannt werden.

Sie können auf der Kontoverwaltungsseite auf der Registerkarte Instances oder mithilfe des Befehls überprüfen, wann eine EC2-Instance zuletzt auf Sicherheitslücken gescannt wurde. ListCoverage

Den Scanmodus verwalten

Ihr EC2-Scanmodus bestimmt, welche Scanmethoden Amazon Inspector bei der Durchführung von EC2-Scans in Ihrem Konto verwendet. Sie können den Scanmodus für Ihr Konto auf der Seite mit den EC2-Scaneinstellungen unter Allgemeine Einstellungen einsehen. Eigenständige Konten oder

von Amazon Inspector delegierte Administratoren können den Scanmodus ändern. Wenn Sie den Scanmodus als delegierter Administrator von Amazon Inspector festlegen, wird dieser Scanmodus für alle Mitgliedskonten in Ihrer Organisation festgelegt. Amazon Inspector bietet die folgenden Scanmodi:

Agent-based Scannen — In diesem Scanmodus verwendet Amazon Inspector ausschließlich die agentenbasierte Scanmethode, um nach Sicherheitslücken in Paketen zu suchen. Dieser Scanmodus scannt nur SSM-verwaltete Instances in Ihrem Konto, bietet jedoch den Vorteil, dass als Reaktion auf neue CVES oder Änderungen an den Instances kontinuierliche Scans durchgeführt werden. Agent-based Das Scannen bietet Amazon Inspector auch eine gründliche Inspektion für berechtigte Instanzen. Dies ist der Standard-Scanmodus für neu aktivierte Konten.

Hybrid-Scan — In diesem Scanmodus verwendet Amazon Inspector eine Kombination aus agentenbasierten und agentenlosen Methoden, um nach Sicherheitslücken in Paketen zu suchen. Für berechtigte EC2-Instances, auf denen der SSM-Agent installiert und konfiguriert ist, verwendet Amazon Inspector die agentenbasierte Methode. Für berechtigte Instances, die nicht über SSM verwaltet werden, verwendet Amazon Inspector die agentenlose Methode für berechtigte EBS-backed Instances.

Um den Scanmodus zu ändern

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mit der AWS-Region Auswahl in der oberen rechten Ecke der Seite die Region aus, in der Sie Ihren EC2-Scanmodus ändern möchten.
3. Wählen Sie im seitlichen Navigationsbereich unter Allgemeine Einstellungen die Option EC2-Scaneinstellungen aus.
4. Wählen Sie unter Scanmodus die Option Bearbeiten aus.
5. Wählen Sie einen Scanmodus und dann Änderungen speichern aus.

Instances von Amazon Inspector-Scans ausschließen

Sie können Windows Instances von Amazon Inspector-Scans ausschließenLinux, indem Sie diese Instances mit dem `InspectorEc2Exclusion` Schlüssel kennzeichnen. Beim Tag-Schlüssel wird nicht zwischen Groß- und Kleinschreibung unterschieden. Die Angabe eines Tag-Werts ist optional. Informationen zum Hinzufügen von Tags finden Sie unter Taggen Ihrer Amazon EC2 EC2-Ressourcen.

Wenn Sie eine Instance für den Ausschluss aus Amazon Inspector-Scans kennzeichnen, markiert Amazon Inspector die Instance als ausgeschlossen und erstellt keine Ergebnisse dafür. Das Amazon Inspector SSM-Plugin wird jedoch weiterhin aufgerufen. Um zu verhindern, dass das Plugin aufgerufen wird, müssen Sie den Zugriff auf Tags in den Instanz-Metadaten zulassen.

> **Note**
>
> Für ausgeschlossene Instanzen werden Ihnen keine Gebühren berechnet.

Darüber hinaus können Sie eine Instance von Scans ohne Agenten ausschließen, indem Sie den AWS KMS Schlüssel, mit dem das Volume verschlüsselt wurde, mit dem Tag kennzeichnen. `InspectorEc2Exclusion` Weitere Informationen finden Sie unter Kennzeichnen von Schlüsseln.

Unterstützte Betriebssysteme

Amazon Inspector scannt unterstützte Mac-, Windows- und Linux-Instances auf Sicherheitslücken in Betriebssystempaketen. Für Linux-Instances kann Amazon Inspector Ergebnisse für Anwendungsprogrammiersprachenpakete erstellen, die verwendet Eingehende Inspektion von Amazon Inspector für Linux-based Amazon EC2 EC2-Instances werden. Für Mac- und Windows-Instances werden nur Betriebssystempakete gescannt.

Informationen zu unterstützten Betriebssystemen, einschließlich der Betriebssysteme, die ohne SSM-Agent gescannt werden können, finden Sie unterStatuswerte Amazon EC2 EC2-Instances.

Eingehende Inspektion von Amazon Inspector für Linux-based Amazon EC2 EC2-Instances

> **Note**
>
> Diese Seite bezieht sich auf Kunden, die sich nicht für Enhanced EC2 Scanning entschieden haben. Kunden, die sich für Enhanced EC2 Scanning entschieden haben, benötigen keine zusätzliche Richtlinienkonfiguration.

Amazon Inspector erweitert den Anwendungsbereich von Amazon EC2-Scans um eine gründliche Inspektion. Mit einer gründlichen Inspektion erkennt Amazon Inspector Paketschwachstellen für Anwendungsprogrammiersprachenpakete in Ihren Linux-based Amazon EC2 EC2-Instances. Amazon Inspector scannt Standardpfade für Programmiersprachen-Paketbibliotheken. Sie können

jedoch zusätzlich zu den Pfaden, die Amazon Inspector standardmäßig scannt, benutzerdefinierte Pfade konfigurieren.

> **Note**
>
> Für eine gründliche Inspektion sind `ssm:PutInventory` `ssm:GetParameter` Genehmigungen erforderlich. Wenn auf der Instance ein IAM-Instance-Profil konfiguriert ist, verwendet Amazon Inspector dieses Profil und ignoriert die DHMC-Rolle. Das Instance-Profil muss diese Berechtigungen enthalten. Wenn kein Instance-Profil festgelegt ist, verwendet Amazon Inspector die konfigurierte Standard-Host-Management-Konfigurationsrolle, die diese Berechtigungen enthalten muss.

Um Deep Inspection-Scans für Ihre Linux-based Amazon EC2 EC2-Instances durchzuführen, verwendet Amazon Inspector Daten, die mit dem Amazon Inspector SSM-Plugin gesammelt wurden. Um das Amazon Inspector SSM-Plug-In zu verwalten und Deep Inspection für Linux durchzuführen, erstellt Amazon Inspector automatisch die SSM-Verknüpfung `InvokeInspectorLinuxSsmPlugin-do-not-delete` in Ihrem Konto. Amazon Inspector sammelt alle 6 Stunden aktualisiertes Anwendungsinventar von Ihren Linux-based Amazon EC2 EC2-Instances.

> **Tiefgehende Inspektion unter Windows und macOS**
>
> Die Tiefeninspektion wird für Windows oder Mac-Instances nicht unterstützt, es sei denn, Sie verwenden Enhanced EC2 Scanning. Bei Enhanced EC2 Scanning wird die Tiefeninspektion vom Amazon Inspector VM Scanner durchgeführt, der macOS InstancesLinux,Windows, und unterstützt.

In diesem Abschnitt wird beschrieben, wie Sie Amazon Inspector Deep Inspection für Amazon EC2 EC2-Instances verwalten, einschließlich der Festlegung benutzerdefinierter Pfade für Amazon Inspector zum Scannen.

Themen

- Auf Deep Inspection zugreifen oder diese deaktivieren
- Benutzerdefinierte Pfade für die Tiefeninspektion mit Amazon Inspector
- Benutzerdefinierte Zeitpläne für die Tiefeninspektion mit Amazon Inspector

- Unterstützte Programmiersprachen

Auf Deep Inspection zugreifen oder diese deaktivieren

> **Note**
>
> Für Konten, die Amazon Inspector nach dem 17. April 2023 aktivieren, wird die Tiefeninspektion automatisch als Teil des Amazon EC2-Scans aktiviert.

Um die Tiefeninspektion zu verwalten

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home
2. Wählen Sie im Navigationsbereich Allgemeine Einstellungen und dann Amazon EC2-Scaneinstellungen aus.
3. Unter Deep Inspection of Amazon EC2 EC2-Instance können Sie benutzerdefinierte Pfade für Ihre Organisation oder für Ihr eigenes Konto festlegen.

Sie können den Aktivierungsstatus für ein einzelnes Konto programmgesteuert mit der API überprüfen. GetEc2DeepInspectionConfiguration Sie können den Aktivierungsstatus programmgesteuert für mehrere Konten mit der API überprüfen. BatchGetMemberEc2DeepInspectionStatus

Wenn Sie Amazon Inspector vor dem 17. April 2023 aktiviert haben, können Sie Deep Inspection über das Konsolenbanner oder die UpdateEc2DeepInspectionConfigurationAPI aktivieren. Wenn Sie der delegierte Administrator für eine Organisation in Amazon Inspector sind, können Sie die BatchUpdateMemberEc2DeepInspectionStatusAPI verwenden, um Deep Inspection für sich und Ihre Mitgliedskonten zu aktivieren.

Sie können Deep Inspection über die UpdateEc2DeepInspectionConfigurationAPI deaktivieren. Mitgliedskonten in einer Organisation können Deep Inspection nicht deaktivieren. Stattdessen muss das Mitgliedskonto von seinem delegierten Administrator mithilfe der BatchUpdateMemberEc2DeepInspectionStatusAPI deaktiviert werden.

Benutzerdefinierte Pfade für die Tiefeninspektion mit Amazon Inspector

Sie können benutzerdefinierte Pfade festlegen, die Amazon Inspector während der Tiefeninspektion Ihrer Amazon EC2 EC2-Instances scannen soll. Wenn Sie einen benutzerdefinierten Pfad festlegen, scannt Amazon Inspector Pakete in diesem Verzeichnis und allen Unterverzeichnissen darin, zusätzlich zu den Speicherorten, die Amazon Inspector standardmäßig scannt.

> **Important**
>
> Benutzerdefinierte Pfade werden fürLinux, und macOS Instances unterstütztWindows, wenn Sie Enhanced EC2 Scanning (den Amazon Inspector VM Scanner) verwenden. Wenn Sie das ältere Amazon Inspector SSM-Plugin anstelle von Enhanced EC2 Scanning verwenden, sind Deep Inspection und benutzerdefinierte Pfade nur für Linux Instances verfügbar.

Alle Konten können bis zu 5 benutzerdefinierte Pfade definieren. Der delegierte Administrator einer Organisation kann 10 benutzerdefinierte Pfade definieren.

Amazon Inspector scannt alle benutzerdefinierten Pfade zusätzlich zu den Speicherorten, die Amazon Inspector standardmäßig für alle Konten scannt. Bei Linux Instances umfassen die Standardspeicherorte Folgendes:

- `/usr/lib`
- `/usr/lib64`
- `/usr/local/lib`
- `/usr/local/lib64`

Auf Windows Instances scannt Amazon Inspector standardmäßig die Speicherorte von Standardbetriebssystemen und Programmiersprachenpaketen auf dem Systemlaufwerk. Software, die an einem benutzerdefinierten oder nicht standardmäßigen Speicherort installiert ist — z. B. auf einem Datenbank- oder Anwendungsserver auf einem Laufwerk wie `D:\` — wird nur gescannt, wenn Sie sie als benutzerdefinierten Pfad hinzufügen.

> **Note**
>
> Benutzerdefinierte Pfade müssen lokale Pfade sein. Amazon Inspector scannt keine zugewiesenen Netzwerkpfade, wie z. B. Netzwerkdateisystem-Mounts oder Amazon S3 S3-Dateisystem-Mounts.

Formatieren benutzerdefinierter Pfade

Ein benutzerdefinierter Pfad darf nicht länger als 256 Zeichen sein. Im Folgenden finden Sie ein Beispiel dafür, wie ein benutzerdefinierter Pfad aussehen könnte:

Beispiel für einen Pfad

```
/home/usr1/project01
```

> **Note**
>
> Das Paketlimit pro Instanz beträgt 5.000. Die maximale Zeit für die Erfassung des Paketinventars beträgt 15 Minuten. Amazon Inspector empfiehlt, benutzerdefinierte Pfade zu wählen, um diese Beschränkungen zu umgehen.

Einen benutzerdefinierten Pfad in der Amazon Inspector Inspector-Konsole und mit der Amazon Inspector Inspector-API einrichten

Die folgenden Verfahren beschreiben, wie Sie einen benutzerdefinierten Pfad für Amazon Inspector Deep Inspection in der Amazon Inspector Inspector-Konsole und mit der Amazon Inspector Inspector-API festlegen. Nachdem Sie einen benutzerdefinierten Pfad festgelegt haben, nimmt Amazon Inspector den Pfad in die nächste Tiefeninspektion auf.

Console

1. Melden Sie sich AWS-Managementkonsole als delegierter Administrator an und öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home
2. Verwenden Sie den AWS-Region Selektor, um die Region auszuwählen, in der Sie das Lambda-Standardscannen aktivieren möchten.

3. Wählen Sie im Navigationsbereich Allgemeine Einstellungen und anschließend EC2-Scaneinstellungen aus.
4. Wählen Sie unter Benutzerdefinierte Pfade für Ihr eigenes Konto die Option Bearbeiten aus.
5. Geben Sie in den Pfad-Textfeldern Ihre benutzerdefinierten Pfade ein.
6. Wählen Sie Speichern.

API

Führen Sie den Befehl UpdateEc2DeepInspectionConfiguration aus. `packagePaths`Geben Sie ein Array von Pfaden an, die gescannt werden sollen.

Benutzerdefinierte Zeitpläne für die Tiefeninspektion mit Amazon Inspector

Standardmäßig erfasst Amazon Inspector alle 6 Stunden ein Anwendungsinventar von Amazon EC2 EC2-Instances. Sie können jedoch die folgenden Befehle ausführen, um zu steuern, wie oft Amazon Inspector dies tut.

Beispielbefehl 1: Zuordnungen auflisten, um die Zuordnungs-ID und das aktuelle Intervall anzuzeigen

Der folgende Befehl zeigt die Zuordnungs-ID für die Assoziation an`InvokeInspectorLinuxSsmPlugin-do-not-delete`.

```
aws ssm list-associations \
--association-filter-list "key=AssociationName,value=InvokeInspectorLinuxSsmPlugin-do-not-delete" \
--region your-Region
```

Beispielbefehl 2: Die Zuordnung so aktualisieren, dass sie ein neues Intervall einschließt

Der folgende Befehl verwendet die Zuordnungs-ID für die Zuordnung`InvokeInspectorLinuxSsmPlugin-do-not-delete`. Sie können die Rate für einen Zeitraum `schedule-expression` zwischen 6 Stunden und einem neuen Intervall, z. B. 12 Stunden, festlegen.

```
aws ssm update-association \
--association-id "your-association-ID" \
--association-name "InvokeInspectorLinuxSsmPlugin-do-not-delete" \
--schedule-expression "rate(6 hours)" \
--region your-Region
```

> **Note**
>
> Je nach Anwendungsfall können Sie das tägliche SSM-Inventarlimit überschreiten, wenn Sie den Tarif für `schedule-expression` einen Tarif zwischen 6 Stunden und einem Intervall wie 30 Minuten festlegen. Dies führt zu verzögerten Ergebnissen, und es kann vorkommen, dass Amazon EC2 EC2-Instances mit teilweisem Fehlerstatus auftreten.

Unterstützte Programmiersprachen

Bei Linux-Instances kann Amazon Inspector Deep Inspector Ergebnisse für Anwendungsprogrammiersprachenpakete und Betriebssystempakete liefern.

Bei Mac- und Windows-Instances kann Amazon Inspector Deep Inspector nur Ergebnisse für Betriebssystempakete liefern.

Weitere Informationen zu unterstützten Programmiersprachen finden Sie unter Unterstützte Programmiersprachen: Amazon EC2 Deep Inspection.

Scanning Windows EC2-Instances mit Amazon Inspector

> **Note**
>
> Diese Seite bezieht sich auf Kunden, die sich nicht für Enhanced EC2 Scanning entschieden haben.

Amazon Inspector erkennt automatisch alle unterstützten Windows Instances und nimmt sie ohne zusätzliche Aktionen in kontinuierliche Scans auf. Informationen darüber, welche Instances unterstützt werden, finden Sie unter Von Amazon Inspector unterstützte Betriebssysteme und Programmiersprachen. Amazon Inspector führt in regelmäßigen Abständen Windows Scans durch. WindowsInstances werden bei Entdeckung und dann alle 6 Stunden gescannt. Sie können das Standard-Scan-Intervall jedoch nach dem ersten Scan anpassen.

Wenn das Amazon EC2-Scannen aktiviert ist, erstellt Amazon Inspector die folgenden SSM-Verknüpfungen für Ihre Windows Ressourcen: `InspectorDistributor-do-not-deleteInspectorInventoryCollection-do-not-delete`, und. `InvokeInspectorSsmPlugin-do-not-delete` Um das Amazon Inspector SSM-Plugin auf Ihren Windows Instances zu installieren, verwendet die `InspectorDistributor-do-not-delete`

SSM-Zuordnung das `AWS-ConfigureAWSPackage`SSM-Dokument und das `AmazonInspector2-InspectorSsmPluginSSM` Distributor-Paket. Weitere Informationen finden Sie unter Das Amazon Inspector SSM-Plugin für Windows. Um Instance-Daten zu sammeln und Amazon Inspector-Ergebnisse zu generieren, führt die `InvokeInspectorSsmPlugin-do-not-delete` SSM-Vereinigung das Amazon Inspector SSM-Plugin in Intervallen von 6 Stunden aus. Sie können diese Einstellung jedoch mithilfe eines Cron- oder Rate-Ausdrucks anpassen.

> **Note**
>
> Amazon Inspector stellt aktualisierte OVAL-Definitionsdateien (Open Vulnerability and Assessment Language) im S3-Bucket bereit`inspector2-oval-prod-`*`your-AWS-Region`*. Der Amazon S3 S3-Bucket enthält OVAL-Definitionen, die in Scans verwendet werden. Diese OVAL-Definitionen sollten nicht geändert werden. Andernfalls sucht Amazon Inspector bei der Veröffentlichung nicht nach neuen CVEs.

Amazon Inspector-Scananforderungen für Windows -Instances

Um eine Windows Instance zu scannen, setzt Amazon Inspector voraus, dass die Instance die folgenden Kriterien erfüllt:

- Die Instance ist eine von SSM verwaltete Instance. Anweisungen zum Einrichten Ihrer Instanz für das Scannen finden Sie unterDen SSM-Agenten konfigurieren.
- Das Instanzbetriebssystem ist eines der unterstützten Windows Betriebssysteme. Eine vollständige Liste der unterstützten Betriebssysteme finden Sie unterStatuswerte Amazon EC2 EC2-Instances.
- Auf der Instance ist das Amazon Inspector SSM-Plugin installiert. Amazon Inspector installiert bei Entdeckung automatisch das Amazon Inspector SSM-Plugin für verwaltete Instances. Einzelheiten zum Plugin finden Sie im nächsten Thema.

> **Note**
>
> Wenn Ihr Host in einer Amazon VPC ohne ausgehenden Internetzugang läuft, erfordert das Windows Scannen, dass Ihr Host auf regionale Amazon S3 S3-Endpunkte zugreifen kann. Informationen zur Konfiguration eines Amazon S3 S3-Amazon-VPC-Endpunkts finden Sie unter Erstellen eines Gateway-Endpunkts im Amazon Virtual Private Cloud-Benutzerhandbuch. Wenn Ihre Amazon VPC-Endpunktrichtlinie den Zugriff auf externe S3-Buckets einschränkt, müssen Sie ausdrücklich den Zugriff auf den von Amazon Inspector

verwalteten Bucket in Ihrem zulassen AWS-Region , in dem die zur Bewertung Ihrer Instance verwendeten OVAL-Definitionen gespeichert sind. Dieser Bucket hat das folgende Format:. `inspector2-oval-prod-`*`REGION`*

Einstellung benutzerdefinierter Zeitpläne für Windows Instanzscans

Sie können die Zeit zwischen Ihren Windows Amazon EC2 EC2-Instance-Scans anpassen, indem Sie einen Cron-Ausdruck oder einen Rate-Ausdruck für die `InvokeInspectorSsmPlugin-do-not-delete` Zuordnung mithilfe von SSM festlegen. Weitere Informationen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für Systems Manager im AWS Systems Manager Benutzerhandbuch oder verwenden Sie die folgenden Anweisungen.

Wählen Sie eines der folgenden Codebeispiele aus, um die Scan-Taktfrequenz für Windows Instances von der Standardeinstellung 6 Stunden auf 12 Stunden zu ändern, indem Sie entweder einen Rate- oder einen Cron-Ausdruck verwenden.

In den folgenden Beispielen müssen Sie die AssociationIdfür die angegebene Assoziation verwenden. `InvokeInspectorSsmPlugin-do-not-delete` Sie können Ihre abrufen, AssociationIdindem Sie den folgenden AWS CLI Befehl ausführen:

```
$ aws ssm list-associations --association-filter-list
 "key=AssociationName,value=InvokeInspectorSsmPlugin-do-not-delete" --region us-east-1
```

> **Note**
>
> Da AssociationIdes sich um Regional handelt, müssen Sie zunächst für jede ID eine eindeutige ID abrufen AWS-Region. Anschließend können Sie den Befehl ausführen, um die Scanfrequenz in jeder Region zu ändern, in der Sie einen benutzerdefinierten Scan-Zeitplan für Windows Instances festlegen möchten.

Example rate expression

```
$ aws ssm update-association \
--association-id "YourAssociationId" \
--association-name "InvokeInspectorSsmPlugin-do-not-delete" \
--schedule-expression "rate(12 hours)"
```

Example cron expression

```
$ aws ssm update-association \
--association-id "YourAssociationId" \
--association-name "InvokeInspectorSsmPlugin-do-not-delete" \
--schedule-expression "cron(0 0/12 * * ? *)"
```

Scannen von Amazon Elastic Container Registry-Container-Images mit Amazon Inspector

Amazon Inspector scannt Container-Images, die in Amazon Elastic Container Registry gespeichert sind, auf Softwareschwachstellen, um Sicherheitslücken in Paketen zu ermitteln. Wenn Sie das Amazon ECR-Scannen aktivieren, legen Sie Amazon Inspector als bevorzugten Scanservice für Ihre private Registrierung fest.

> **Note**
>
> Amazon ECR verwendet eine Registrierungsrichtlinie, um einem AWS Prinzipal Berechtigungen zu erteilen. Dieser Principal verfügt über die erforderlichen Berechtigungen, um Amazon Inspector Inspector-APIs zum Scannen aufzurufen. Wenn Sie den Geltungsbereich Ihrer Registrierungsrichtlinie festlegen, dürfen Sie die `ecr:*` Aktion nicht hinzufügen oder `PutRegistryScanningConfiguration` eintragen`deny`. Dies führt zu Fehlern auf Registrierungsebene, wenn das Scannen für Amazon ECR aktiviert und deaktiviert wird.

Beim einfachen Scannen können Sie Ihre Repositorys so konfigurieren, dass sie bei Push-Scans scannen oder manuelle Scans durchführen. Mit den erweiterten Scans können Sie auf der Registrierungsebene nach Sicherheitslücken in Betriebssystemen und Programmiersprachenpaketen suchen. Einen direkten Vergleich der Unterschiede zwischen einfachem und erweitertem Scannen finden Sie in den häufig gestellten Fragen zu Amazon Inspector.

> **Note**
>
> Das einfache Scannen wird über Amazon ECR bereitgestellt und abgerechnet. Weitere Informationen finden Sie unter Amazon Elastic Container Registry — Preise. Erweitertes

Scannen wird über Amazon Inspector bereitgestellt und abgerechnet. Weitere Informationen erhalten Sie unter Amazon Inspector: Preise.

Informationen zur Aktivierung des Amazon ECR-Scannens finden Sie unter Einen Scantyp aktivieren. Informationen zum Anzeigen von Ergebnissen finden Sie unter Amazon Inspector Inspector-Ergebnisse anzeigen. Informationen zum Anzeigen von Ergebnissen in Amazon ECR auf Bildebene finden Sie unter Scannen von Bildern im Amazon Elastic Container Registry User Guide. Sie können Ergebnisse verwalten, indem Sie Amazon verwenden, die für Standardscans AWS-Services nicht verfügbar sind EventBridge.AWS Security Hub CSPM

Sie können die Scan-Konfiguration für jedes Repository in Amazon Inspector über Deckungsseiten und APIs einsehen. Die Konfigurationseinstellungen für einfaches Scannen im Vergleich zum kontinuierlichen Scannen können jedoch nur in Amazon ECR geändert werden. Amazon Inspector bietet Einblick in diese Einstellungen, bietet jedoch keine direkten Änderungsmöglichkeiten. Weitere Informationen finden Sie unter Bilder nach Softwareschwachstellen in Amazon ECR scannen im Amazon ECR-Benutzerhandbuch.

Dieser Abschnitt enthält Informationen zum Amazon ECR-Scannen und beschreibt, wie Sie das erweiterte Scannen für Amazon ECR-Repositorys konfigurieren.

Scanverhalten für Amazon ECR-Scans

Wenn Sie das Amazon ECR-Scannen zum ersten Mal aktivieren, erkennt Amazon Inspector Bilder, die innerhalb der letzten 14 Tage übertragen wurden. Amazon Inspector scannt dann die Bilder und setzt den Scanstatus auf`ACTIVE`. Amazon Inspector scannt nur Bilder, die in ECR aktiv sind (`imageStatus`Feld ist`ACTIVE`). Bilder mit dem Status Archiviert in ECR (`imageStatus`Feld ist`ARCHIVED`) werden von Amazon Inspector nicht gescannt.

Wenn kontinuierliches Scannen aktiviert ist, überwacht Amazon Inspector Bilder, sofern sie innerhalb von 14 Tagen übertragen wurden (standardmäßig), das Datum der letzten Verwendung innerhalb von 14 Tagen liegt (standardmäßig) oder die Bilder innerhalb der konfigurierten Dauer für das erneute Scannen gescannt werden. Für Amazon Inspector Inspector-Konten, die vor dem 16. Mai 2024 erstellt wurden, ist die Standardkonfiguration ein erneutes Scannen, um Bilder zu überwachen, wenn sie innerhalb der letzten 90 Tage per Push oder Pull übertragen wurden. Weitere Informationen finden Sie unter Konfiguration der Dauer des Amazon ECR-Neuscans.

Für kontinuierliches Scannen initiiert Amazon Inspector in den folgenden Situationen neue Schwachstellenscans von Container-Images:

- Immer wenn ein neues Container-Image übertragen wird.
- Immer wenn Amazon Inspector seiner Datenbank ein neues CVE-Element (Common Vulnerabilities and Exposures) hinzufügt und dieses CVE für dieses Container-Image relevant ist (nur kontinuierliches Scannen).
- Immer wenn ein Container-Image in ECR von archiviert auf aktiv umgestellt wird.

Wenn Sie Ihr Repository für On-Push-Scannen konfigurieren, werden Bilder nur gescannt, wenn Sie sie per Push übertragen.

Sie können im Tab Container-Images auf der Kontoverwaltungsseite oder mithilfe der ListCoverageAPI überprüfen, wann ein Container-Image zuletzt auf Sicherheitslücken überprüft wurde. Amazon Inspector aktualisiert das Feld Zuletzt gescannt am eines Amazon ECR-Bilds als Reaktion auf die folgenden Ereignisse:

- Wenn Amazon Inspector einen ersten Scan eines Container-Images abschließt.
- Wenn Amazon Inspector ein Container-Image erneut scannt, weil ein neues CVE-Element (Common Vulnerabilities and Exposures), das sich auf dieses Container-Image auswirkt, zur Amazon Inspector Inspector-Datenbank hinzugefügt wurde.

Archivierte ECR-Container-Bilder

Amazon Inspector scannt keine Container-Bilder, die in ECR (`imageStatusisARCHIVED`) archiviert wurden. Wenn ein aktives Bild in ECR auf archiviert umgestellt wird, schließt Amazon Inspector die Ergebnisse automatisch und löscht die Ergebnisse dann nach 3 Tagen. Wenn ein archiviertes Container-Image in ECR auf aktiv umgestellt wird, löst Amazon Inspector einen neuen Scan aus.

Zuordnung von Container-Images zu laufenden Containern

Amazon Inspector bietet ein umfassendes Container-Sicherheitsmanagement, indem Container-Images laufenden Containern in Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS) zugeordnet werden. Diese Zuordnungen bieten Einblicke in Sicherheitslücken bei Bildern auf laufenden Containern.

> **Note**
>
> Die verwaltete Richtlinie `AWSReadOnlyAccess` allein bietet keine ausreichenden Berechtigungen, um die Zuordnung zwischen Amazon ECR-Images und laufenden

Containern anzuzeigen. Sie benötigen `AWSReadOnlyAccess` sowohl die als auch die `AWSInspector2ReadOnlyAccess` verwalteten Richtlinien, um Informationen zur Zuordnung von Container-Images anzuzeigen.

Sie können Abhilfemaßnahmen auf der Grundlage betrieblicher Risiken priorisieren und den Sicherheitsschutz im gesamten Container-Ökosystem aufrechterhalten. Sie können sehen, wie viele Container-Images derzeit verwendet werden und welche Container-Images in den letzten 24 Stunden zuletzt auf einem Amazon ECS- oder Amazon EKS-Cluster verwendet wurden. Sie können auch sehen, wie viele Amazon ECS-Aufgaben und Amazon EKS-Pods bereitgestellt werden. Diese Informationen finden Sie in der Amazon Inspector Inspector-Konsole auf dem Detailbildschirm für die Ergebnisse von Container-Images `ecrImageInUseCount` und mit den `ecrImageLastInUseAt` Filtern und für den `FilterCriteria`Datentyp. Bei neuen Container-Images oder Konten kann es bis zu 36 Stunden dauern, bis Daten verfügbar sind. Danach werden diese Daten alle 24 Stunden aktualisiert. Weitere Informationen finden Sie unter Amazon Inspector Inspector-Ergebnisse anzeigen und Details zu Amazon Inspector Inspector-Ergebnissen anzeigen.

> **Note**
>
> Diese Daten werden automatisch an Amazon ECR Findings gesendet, wenn Sie das Amazon ECR-Scannen aktivieren und Ihr Repository für kontinuierliches Scannen konfigurieren. Kontinuierliches Scannen muss auf Amazon ECR-Repository-Ebene konfiguriert werden. Weitere Informationen finden Sie unter Verbessertes Scannen im Amazon Elastic Container Registry User Guide.

Sie können Container-Images auch anhand des Datums der letzten Verwendung aus Clustern erneut scannen.

Diese Funktion wird auch auf Fargate mit Amazon ECS und Amazon EKS unterstützt.

Unterstützte Betriebssysteme und Medientypen

Informationen zu unterstützten Betriebssystemen finden Sie unterUnterstützte Betriebssysteme: Amazon ECR-Scannen mit Amazon Inspector.

Amazon Inspector-Scans von Amazon ECR-Repositorys decken die folgenden unterstützten Medientypen ab:

Image-Manifest

- `"application/vnd.oci.image.manifest.v1+json"`
- `"application/vnd.docker.distribution.manifest.v2+json"`

Image-Konfiguration

- `"application/vnd.docker.container.image.v1+json"`
- `"application/vnd.oci.image.config.v1+json"`

Bildebenen

- `"application/vnd.docker.image.rootfs.diff.tar"`
- `"application/vnd.docker.image.rootfs.diff.tar.gzip"`
- `"application/vnd.docker.image.rootfs.foreign.diff.tar.gzip"`
- `"application/vnd.oci.image.layer.v1.tar"`
- `"application/vnd.oci.image.layer.v1.tar+gzip"`
- `"application/vnd.oci.image.layer.v1.tar+zstd"`
- `"application/vnd.oci.image.layer.nondistributable.v1.tar"`
- `"application/vnd.oci.image.layer.nondistributable.v1.tar+gzip"`

> **Note**
>
> Amazon Inspector unterstützt den `"application/vnd.docker.distribution.manifest.list.v2+json"` Medientyp für das Scannen von Amazon ECR-Repositorys nicht.

Konfiguration der Dauer des erneuten Scans von Amazon ECR

Die Einstellung für die Dauer des erneuten Scans in Amazon ECR bestimmt, wie lange Amazon Inspector kontinuierlich Container-Images in Repositorys überwacht. Sie konfigurieren die Dauer des erneuten Scans für das Datum der letzten Verwendung des Bilds, das Datum des letzten Abrufs und das Push-Datum. Es hat sich bewährt, die Dauer des erneuten Scans so zu konfigurieren, dass sie am besten zu Ihrer Umgebung passt.

Wenn Sie häufig Images erstellen, wählen Sie eine kürzere Scandauer. Wählen Sie für Bilder, die über einen längeren Zeitraum verwendet werden, eine längere Scandauer. Die Standardscandauer für neue Konten, einschließlich neuer Konten, die zu einer Organisation hinzugefügt wurden, beträgt 14 Tage.

Amazon Inspector überwacht weiterhin ein Bild und scannt es erneut, solange es zuletzt in einem Cluster verwendet oder innerhalb von 14 Tagen übertragen wurde (standardmäßig). Wenn ein Bild innerhalb des konfigurierten Push- und letzten Nutzungsdatums nicht per Push übertragen oder zuletzt in einem laufenden Container verwendet wurde, beendet Amazon Inspector die Überwachung. Es besteht die Möglichkeit, die Einstellung so zu ändern, dass Bilder bei Bedarf nach dem Datum des letzten Abrufs statt nach dem Datum der letzten Verwendung überwacht werden. Wenn Amazon Inspector die Überwachung eines Bilds beendet, wird der Statuscode für den Bildscan auf inaktiv und der Ursachencode auf abgelaufen gesetzt. Amazon Inspector plant dann, dass alle zugehörigen Bildergebnisse geschlossen werden.

Wenn Sie die Dauer des Push-Datums verlängern, wendet Amazon Inspector die Änderung auf alle aktiv gescannten Bilder in Repositorys an, die für kontinuierliches Scannen konfiguriert sind. Inaktive Bilder bleiben jedoch inaktiv, auch wenn Sie sie innerhalb der neuen Dauer per Push übertragen.

Wenn Sie die Dauer des erneuten Scans von einem delegierten Administratorkonto aus konfigurieren, wendet Amazon Inspector die Einstellung auf alle Mitgliedskonten in der Organisation an. Wenn das delegierte Administratorkonto das Amazon ECR-Scannen nicht aktiviert, kann es keine Cluster für ein API-Image anzeigen.

Bei Images mit mehreren Architekturen wird die Nachverfolgung des Datums der letzten Verwendung nicht unterstützt. Bei der Verwendung von Images mit mehreren Architekturen empfehlen wir, das Scannen auf der Grundlage von Pull- oder Push-Ereignissen für Bilder zu konfigurieren und nicht auf dem Datum der letzten Verwendung, um ein korrektes Verhalten beim erneuten Scannen zu gewährleisten.

> **Note**
>
> Alle Einstellungen für die Dauer des erneuten Scans, die vor dem 16. Mai 2024 konfiguriert wurden, bleiben unverändert. Sie können weiterhin alle zuvor konfigurierten Standardeinstellungen verwenden.

Dauer des erneuten Scans von Bildern

Die Dauer des erneuten Scans von Bildern bestimmt, wie lange Amazon Inspector Bilder überwacht. Die Dauer des erneuten Scannens von Bildern umfasst zwei Modi: Datum der letzten Verwendung (Standard) oder Datum des letzten Abrufs. Wählen Sie Datum der letzten Verwendung (Standard), wenn Sie das Datum der letzten Verwendung aus Ihrer Amazon ECS/Amazon EKS-Cluster-Aktivität verwenden möchten. Wählen Sie Letztes Abrufdatum, wenn Sie das Datum des letzten Abrufs aus Ihren Amazon ECR-Bildern verwenden möchten, um Bilder erneut zu scannen. Die folgenden Optionen sind für die Dauer des erneuten Scans verfügbar:

- 3 Tage
- 7 Tage
- 14 Tage (Standard)
- 30 Tage
- 60 Tage
- 90 Tage
- 180 Tage

Dauer des Bild-Push-Datums

Die Dauer des Image-Push-Datums bestimmt, wie lange Amazon Inspector Bilder kontinuierlich überwacht, nachdem sie in Repositorys übertragen wurden. Die folgenden Optionen sind für die Dauer des erneuten Scans verfügbar:

- 3 Tage
- 7 Tage
- 14 Tage (Standard)
- 30 Tage
- 60 Tage
- 90 Tage
- 180 Tage
- Nutzungsdauer

So konfigurieren Sie die Dauer des erneuten Scans von Amazon ECR

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.

2. Wählen Sie den AWS-Region Ort aus, an dem Sie die Dauer des Amazon ECR-Neuscans konfigurieren möchten.
3. Wählen Sie im Navigationsbereich Allgemeine Einstellungen und dann ECR-Scaneinstellungen aus.
4. Wählen Sie unter Dauer des erneuten ECR-Scans den Modus für das erneute Scannen von Bildern und dann die entsprechende Dauer aus.
5. Wählen Sie unter Bild-Push-Datum das Bild-Push-Datum aus.
6. Wählen Sie Speichern.

Grundlegendes zum Status von ECR-Container-Images

Inspector scannt nur `ACTIVE` Bilder in ECR-Container-Bildern. ECR-Container-Bilder mit einem `ARCHIVED` Status werden nicht gescannt. Weitere Informationen zum Scanverhalten finden Sie unterScanverhalten für Amazon ECR-Scans.

Wenn der Bildstatus eines ECR-Container-Images in ECR zu wechselt, verwendet Inspector das `lastActivatedAt` Feld`ACTIVE`, um die Dauer des erneuten Scannens zu überwachen.

Scanning AWS Lambda funktioniert mit Amazon Inspector

Die Unterstützung von AWS Lambda Funktionen und Ebenen durch Amazon Inspector ermöglicht kontinuierliche automatisierte Bewertungen von Sicherheitslücken. Amazon Inspector bietet zwei Arten von Lambda-Funktionsscans:

Standard-Scanning mit Amazon Inspector Lambda

Dieser Scantyp ist der standardmäßige Lambda-Scantyp. Es scannt Anwendungsabhängigkeiten in Lambda-Funktionen und -Layern auf Paketschwachstellen.

Scannen von Lambda-Code mit Amazon Inspector

Dieser Scantyp scannt benutzerdefinierten Anwendungscode in Ihren Lambda-Funktionen und -Layern auf Code-Schwachstellen. Sie können das Lambda-Standardscannen oder das Lambda-Standardscannen mit Lambda-Code-Scan aktivieren.

Wenn Sie das Lambda-Code-Scannen aktivieren möchten, müssen Sie zuerst das Lambda-Standardscannen aktivieren. Weitere Informationen finden Sie unter Einen Scan-Typ aktivieren.

Wenn Sie den Lambda-Funktionsscan aktivieren, erstellt Amazon Inspector die folgenden serviceverknüpften Kanäle in Ihrem Konto: `cloudtrail:CreateServiceLinkedChannel` und. `cloudtrail:DeleteServiceLinkedChannel` Amazon Inspector verwaltet diese Kanäle und verwendet sie, um CloudTrail Ereignisse für Scans zu überwachen. Die Kanäle ermöglichen es Ihnen, CloudTrail Ereignisse in Ihrem Konto so zu verfolgen, als ob Sie eine Spur davon gehabt hätten CloudTrail. Wir empfehlen, deinen eigenen Trail zu erstellen CloudTrail , um Ereignisse in deinem Konto zu verwalten. Informationen zur Anzeige dieser Kanäle findest du im AWS CloudTrail Benutzerhandbuch unter Mit Diensten verknüpfte Kanäle anzeigen.

> **Note**
>
> Amazon Inspector unterstützt das Scannen von Lambda-Funktionen, die mit vom Kunden verwalteten Schlüsseln verschlüsselt sind, nicht. Dies gilt für Lambda-Standardscans und Lambda-Code-Scans.

Scanverhalten beim Scannen mit Lambda-Funktionen

Nach der Aktivierung scannt Amazon Inspector alle Lambda-Funktionen, die in den letzten 90 Tagen in Ihrem Konto aufgerufen oder aktualisiert wurden. Amazon Inspector initiiert Schwachstellenscans von Lambda-Funktionen in den folgenden Situationen:

- Sobald Amazon Inspector eine bestehende Lambda-Funktion entdeckt.
- Wenn Sie eine neue Lambda-Funktion für den Lambda-Service bereitstellen.
- Wenn Sie ein Update für den Anwendungscode oder die Abhängigkeiten einer vorhandenen Lambda-Funktion oder ihrer Layer bereitstellen.
- Immer wenn Amazon Inspector seiner Datenbank ein neues CVE-Element (Common Vulnerabilities and Exposures) hinzufügt und dieses CVE für Ihre Funktion relevant ist.

Amazon Inspector überwacht jede Lambda-Funktion während ihrer gesamten Lebensdauer, bis sie entweder gelöscht oder vom Scannen ausgeschlossen wird.

Auf der Registerkarte Lambda-Funktionen auf der Kontoverwaltungsseite oder mithilfe der API können Sie überprüfen, wann eine Lambda-Funktion zuletzt auf Sicherheitslücken überprüft wurde. ListCoverage Amazon Inspector aktualisiert das Feld Zuletzt gescannt am für eine Lambda-Funktion als Reaktion auf die folgenden Ereignisse:

- Wenn Amazon Inspector einen ersten Scan einer Lambda-Funktion abschließt.
- Wenn eine Lambda-Funktion aktualisiert wird.
- Wenn Amazon Inspector eine Lambda-Funktion erneut scannt, weil ein neues CVE-Element, das sich auf diese Funktion auswirkt, zur Amazon Inspector Inspector-Datenbank hinzugefügt wurde.

Unterstützte Laufzeiten und geeignete Funktionen

Amazon Inspector unterstützt unterschiedliche Laufzeiten für Lambda-Standardscans und Lambda-Code-Scans. Eine Liste der unterstützten Laufzeiten für jeden Scan-Typ finden Sie unter und. Unterstützte Laufzeiten: Amazon Inspector Lambda Standard-Scanning Unterstützte Laufzeiten: Amazon Inspector Lambda-Code-Scanning

Eine Lambda-Funktion muss nicht nur über eine unterstützte Laufzeit verfügen, sondern auch die folgenden Kriterien erfüllen, um für Amazon Inspector-Scans in Frage zu kommen:

- Die Funktion wurde in den letzten 90 Tagen aufgerufen oder aktualisiert.
- Die Funktion ist markiert`$LATEST`.
- Die Funktion ist nicht von Scans nach Tags ausgeschlossen.

> **Note**
>
> Lambda-Funktionen, die in den letzten 90 Tagen nicht aufgerufen oder geändert wurden, werden automatisch von Scans ausgeschlossen. Amazon Inspector setzt das Scannen einer automatisch ausgeschlossenen Funktion fort, wenn sie erneut aufgerufen wird oder wenn Änderungen am Lambda-Funktionscode vorgenommen werden.

Standard-Scanning mit Amazon Inspector Lambda

Das Standard-Scannen von Amazon Inspector Lambda identifiziert Softwareschwachstellen in den Abhängigkeiten von Anwendungspaketen, die Sie Ihrem Lambda-Funktionscode und den Lambda-Funktionsschichten hinzufügen. Wenn Ihre Lambda-Funktion beispielsweise eine Version des `python-jwt` Pakets mit einer bekannten Sicherheitslücke verwendet, generiert der Lambda-Standardscan einen Befund für diese Funktion.

Wenn Amazon Inspector eine Sicherheitslücke in den Abhängigkeiten Ihrer Lambda-Funktionsanwendung feststellt, erstellt Amazon Inspector eine detaillierte Suche nach dem Typ der Sicherheitslücke in Paketen.

Anweisungen zur Aktivierung eines Scan-Typs finden Sie unterEinen Scantyp aktivieren.

> **Note**
>
> Das Lambda-Standardscannen scannt nicht die AWS SDK-Abhängigkeit, die standardmäßig in der Lambda-Laufzeitumgebung installiert ist. Amazon Inspector scannt nur Abhängigkeiten, die mit dem Funktionscode hochgeladen oder von einer Ebene übernommen wurden.

> **Note**
>
> Wenn Sie das Standardscannen von Amazon Inspector Lambda deaktivieren, wird auch das Scannen von Amazon Inspector Lambda-Code deaktiviert.

Funktionen vom Lambda-Standardscan ausschließen

Sie können Lambda-Funktionen Tags hinzufügen, sodass Sie sie von Amazon Inspector Lambda-Standardscans ausschließen können. Das Ausschließen von Funktionen aus Scans kann verhindern, dass Warnmeldungen nicht bearbeitet werden können. Wenn Sie eine Funktion für den Ausschluss kennzeichnen, muss das Tag das folgende Schlüssel-Wert-Paar haben.

- Schlüssel: `InspectorExclusion`
- Wert: `LambdaStandardScanning`

In diesem Thema wird beschrieben, wie Sie eine Funktion kennzeichnen, sodass sie von Scans ausgeschlossen wird. Weitere Informationen zum Hinzufügen von Tags in Lambda finden Sie unter Verwenden von Tags in Lambda-Funktionen.

So schließen Sie eine Funktion von Scans aus

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Lambda-Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im Navigationsbereich Funktionen aus.

3. Wählen Sie den Namen der Funktion, die Sie von Amazon Inspector Lambda-Standardscans ausschließen möchten.
4. Wählen Sie Configuraton (Konfiguration) und dann Tags aus.
5. Wählen Sie Tags verwalten und dann Neues Tag hinzufügen.
   a. Geben Sie für Key (Schlüssel) `InspectorExclusion` ein.
   b. Geben Sie für Wert `LambdaStandardScanning` ein
6. Wählen Sie Save (Speichern) aus.

Scannen von Lambda-Code mit Amazon Inspector

> **Important**
>
> Diese Funktion erfasst Ausschnitte von Lambda-Funktionen, um erkannte Sicherheitslücken hervorzuheben. Diese Snippets können fest codierte Anmeldeinformationen und andere vertrauliche Materialien enthalten.

Mit dieser Funktion scannt Amazon Inspector Anwendungscode in einer Lambda-Funktion auf Codeschwachstellen, die auf bewährten AWS Sicherheitsmethoden basieren, um Datenlecks, Injektionsfehler, fehlende Verschlüsselung und schwache Kryptografie zu erkennen. Amazon Inspector verwendet automatisiertes Denken und maschinelles Lernen, um Ihren Anwendungscode für Lambda-Funktionen zu bewerten. Es verwendet auch interne Detektoren, die in Zusammenarbeit mit Amazon Q entwickelt wurden, um Richtlinienverstöße und Sicherheitslücken zu identifizieren.

Amazon Inspector generiert eine Code-Schwachstelle, wenn es eine Sicherheitslücke im Anwendungscode Ihrer Lambda-Funktion entdeckt. Dieser Erkennungstyp umfasst einen Codeausschnitt, der das Problem zeigt und wo Sie das Problem in Ihrem Code finden können. Außerdem wird vorgeschlagen, wie das Problem behoben werden kann. Der Vorschlag umfasst plug-and-play Codeblöcke, mit denen Sie anfällige Codezeilen ersetzen können. Diese Codekorrekturen werden zusätzlich zu den allgemeinen Anleitungen zur Codebehebung für diesen Befundtyp bereitgestellt.

Vorschläge zur Codekorrektur basieren auf automatisierter Argumentation. Einige Vorschläge zur Codekorrektur funktionieren möglicherweise nicht wie beabsichtigt. Sie sind für die Vorschläge zur Codekorrektur verantwortlich, die Sie übernehmen. Lesen Sie sich die Vorschläge zur Codekorrektur immer durch, bevor Sie sie übernehmen. Möglicherweise müssen Sie sie bearbeiten,

um sicherzustellen, dass Ihr Code wie vorgesehen funktioniert. Weitere Informationen finden Sie in der Richtlinie für verantwortungsvolle KI.

Wenn Sie das Lambda-Code-Scannen aktivieren möchten, müssen Sie zuerst das Lambda-Standardscannen aktivieren. Weitere Informationen finden Sie unter Einen Scantyp aktivieren. Informationen darüber, welche diese Funktion AWS-Regionen unterstützen, finden Sie unterRegion-specific Verfügbarkeit von Funktionen.

Verschlüsseln Ihres Codes im Code — Sicherheitslücken

Amazon Q speichert Codefragmente, bei denen festgestellt wurde, dass sie im Zusammenhang mit einer Code-Schwachstelle stehen, die mithilfe von Lambda-Code-Scans gefunden wurde. Standardmäßig kontrolliert Amazon Q den AWS eigenen Schlüssel, der zur Verschlüsselung Ihres Codes verwendet wird. Sie können jedoch Ihren eigenen, vom Kunden verwalteten Schlüssel für die Verschlüsselung über die Amazon Inspector API verwenden. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand für den Code in Ihren Ergebnissen.

Funktionen vom Lambda-Code-Scannen ausschließen

Sie können Lambda-Funktionen Tags hinzufügen, sodass Sie sie von Amazon Inspector Lambda-Codescans ausschließen können. Wenn Sie Funktionen von Scans ausschließen, können Sie verhindern, dass Warnmeldungen nicht bearbeitet werden können. Wenn Sie eine Funktion für den Ausschluss kennzeichnen, muss das Tag das folgende Schlüssel-Wert-Paar haben.

- Schlüssel: `InspectorCodeExclusion`
- Wert — `LambdaCodeScanning`

In diesem Thema wird beschrieben, wie Sie eine Funktion kennzeichnen, sodass sie von Codescans ausgeschlossen wird. Weitere Informationen zum Hinzufügen von Tags in Lambda finden Sie unter Verwenden von Tags in Lambda-Funktionen.

So schließen Sie eine Funktion von Codescans aus

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Lambda-Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im Navigationsbereich Funktionen aus.
3. Wählen Sie den Namen der Funktion, die Sie von Amazon Inspector Lambda-Codescans ausschließen möchten.

4. Wählen Sie Configuraton (Konfiguration) und dann Tags aus.
5. Wählen Sie „Tags verwalten“ und dann „Neues Tag hinzufügen“.
    a. Geben Sie für Key (Schlüssel) `InspectorCodeExclusion` ein.
    b. Geben Sie für Wert `LambdaCodeScanning` ein
6. Wählen Sie Save (Speichern) aus.

Deaktivieren eines Scantyps in Amazon Inspector

Wenn Sie einen Scantyp deaktivieren, verlieren Sie den Zugriff auf alle Ergebnisse, die der Scantyp erbracht hat. Wenn Sie den Scan-Typ reaktivieren, scannt Amazon Inspector alle infrage kommenden Ressourcen, um neue Ergebnisse zu generieren. Wenn Sie Ihre Ergebnisse aufzeichnen möchten, können Sie sie als Ergebnisbericht in einen Amazon Simple Storage Service (Amazon S3) -Bucket exportieren. Weitere Informationen finden Sie unter Amazon Inspector-Ergebnisberichte exportieren. Wenn Sie einen Scan-Typ deaktivieren, können in dem AWS Konto, in dem Sie den Scan-Typ deaktiviert haben, die folgenden Änderungen auftreten:

Amazon EC2-Scannen

Wenn Sie das Scannen von Amazon Inspector Amazon EC2 nach einem Konto deaktivieren, werden die folgenden SSM-Verknüpfungen gelöscht:

- `InspectorDistributor-do-not-delete`
- `InspectorInventoryCollection-do-not-delete`
- `InspectorLinuxDistributor-do-not-delete`
- `InvokeInspectorLinuxSsmPlugin-do-not-delete`
- `InvokeInspectorSsmPlugin-do-not-delete`.

Darüber hinaus wird das Amazon Inspector SSM-Plugin von allen Windows Hosts entfernt. Weitere Informationen finden Sie unter EC2-Instance wird Windows gescannt.

Amazon ECR-Scannen

Wenn Sie das Amazon ECR-Scannen für ein Konto deaktivieren, ändert sich der Kontotyp Amazon ECR von Erweitertes Scannen mit Amazon Inspector zu Standard-Scannen mit Amazon ECR.

Lambda-Standardscannen

Wenn Sie den Lambda-Standardscan für ein Konto deaktivieren, deaktivieren Sie den Lambda-Code-Scan, wenn der Scantyp aktiviert war. Sie löschen auch den CloudTrail serviceverknüpften Kanal, den Amazon Inspector erstellt, wenn Sie das Lambda-Standardscannen aktivieren.

Amazon Inspector Codesicherheit

Wenn Sie Code Security für Ihr Konto deaktivieren, löschen Sie alle damit verbundenen Integrationen, Projekte und Scankonfigurationen. Wenn Ihr Konto der delegierte Administrator für eine Organisation ist, deaktivieren Sie Code Security nur für Ihr Konto, und Mitgliedskonten werden zu eigenständigen Konten.

Scans deaktivieren

Wenn Sie alle Scanarten für ein Konto deaktivieren, wird Amazon Inspector für dieses Konto in diesem Konto deaktiviert. AWS-Region Weitere Informationen finden Sie unter Amazon Inspector deaktivieren.

Um dieses Verfahren für eine Umgebung mit mehreren Konten abzuschließen, folgen Sie diesen Schritten, während Sie als delegierter Amazon Inspector-Administrator angemeldet sind.

Console

Um Scans zu deaktivieren

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mithilfe der AWS-Region Auswahl in der oberen rechten Ecke der Seite die Region aus, in der Sie Scans deaktivieren möchten.
3. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus.
4. Wählen Sie die Registerkarte Konten, um den Scanstatus eines Kontos anzuzeigen.
5. Aktivieren Sie das Kontrollkästchen jedes Kontos, für das Sie Scans deaktivieren möchten.
6. Wählen Sie Aktionen und dann unter den Deaktivierungsoptionen den Scantyp aus, den Sie deaktivieren möchten.
7. (Empfohlen) Wiederholen Sie diese Schritte AWS-Region für jeden Scantyp, für den Sie diesen Scantyp deaktivieren möchten.

API

Führen Sie den Vorgang „API deaktivieren“ aus. Geben Sie in der Anfrage die Konto-IDs an, für die Sie Scans deaktivieren möchten, und `resourceTypes` geben Sie eine oder mehrere der,, oder an, an, an`EC2`, an, an`ECR`, an`LAMBDA`, um Scans `LAMBDA_CODE` zu deaktivieren.

Center for Internet Security (CIS) scannt nach Amazon EC2 EC2-Instance-Betriebssystemen

Amazon Inspector CIS-Scans (CIS-Scans) vergleichen Ihre Amazon EC2 EC2-Instance-Betriebssysteme, um sicherzustellen, dass Sie sie gemäß den vom Center for Internet Security festgelegten Best-Practice-Empfehlungen konfiguriert haben. CIS Security Benchmarks bietet branchenübliche Konfigurationsgrundlagen und Best Practices für die sichere Konfiguration eines Systems. Sie können CIS-Scans durchführen oder planen, nachdem Sie Amazon Inspector EC2-Scans für ein Konto aktiviert haben. Informationen zur Aktivierung des Amazon EC2-Scannens finden Sie unter Aktivieren eines Scantyps.

> **Note**
>
> CIS-Standards sind für x86_64-Betriebssysteme vorgesehen. Manche Prüfungen werden möglicherweise nicht ausgewertet oder geben ungültige Anweisungen zur Behebung von Ressourcen zurück. ARM-based

Amazon Inspector führt CIS-Scans auf Amazon EC2 EC2-Zielinstanzen auf der Grundlage von Instance-Tags und Ihrem definierten Scan-Zeitplan durch. Amazon Inspector führt für jede Ziel-Instance eine Reihe von Instance-Prüfungen durch. Bei jeder Prüfung wird bewertet, ob Ihre Systemkonfiguration den spezifischen CIS-Benchmark-Empfehlungen entspricht. Jeder Scheck hat eine CIS-Scheck-ID und einen Titel, die einer CIS-Benchmark-Empfehlung für diese Plattform entsprechen. Wenn ein CIS-Scan abgeschlossen ist, können Sie anhand der Ergebnisse sehen, welche Instanzprüfungen für dieses System bestanden, übersprungen oder fehlgeschlagen sind.

> **Note**
>
> Um CIS-Scans durchführen oder planen zu können, benötigen Sie eine sichere Internetverbindung. Wenn Sie jedoch CIS-Scans auf privaten Instances ausführen möchten, müssen Sie einen VPC-Endpunkt verwenden.

Themen

- Amazon EC2 EC2-Instance-Anforderungen für Amazon Inspector CIS-Scans

- CIS-Scans ausführen
- Überlegungen zur Verwaltung von Amazon Inspector CIS-Scans mit AWS Organizations
- Amazon Inspector-eigene Amazon S3 S3-Buckets, die für Amazon Inspector CIS-Scans verwendet werden
- Eine CIS-Scankonfiguration erstellen
- CIS-Scanergebnisse anzeigen
- Bearbeitung einer CIS-Scankonfiguration
- CIS-Scanergebnisse herunterladen

Amazon EC2 EC2-Instance-Anforderungen für Amazon Inspector CIS-Scans

Um einen CIS-Scan auf Ihrer Amazon EC2 EC2-Instance auszuführen, muss die Amazon EC2 EC2-Instance die folgenden Kriterien erfüllen:

- Das Instanzbetriebssystem ist eines der unterstützten Betriebssysteme für CIS-Scans. Weitere Informationen finden Sie unter Von Amazon Inspector unterstützte Betriebssysteme und Programmiersprachen.
- Die Instance ist eine Amazon EC2 Systems Manager Manager-Instance. Weitere Informationen finden Sie unter Arbeiten mit dem SSM-Agenten im AWS Systems Manager Benutzerhandbuch.
- Das Amazon Inspector SSM-Plugin ist auf der Instance installiert. Amazon Inspector installiert dieses Plugin automatisch auf verwalteten Instances.
- Die Instance verfügt über ein Instance-Profil, das SSM Berechtigungen zur Verwaltung der Instance und Amazon Inspector zur Ausführung von CIS-Scans für diese Instance gewährt. Um diese Berechtigungen zu gewähren, fügen Sie die AmazonInspector2ManagedCisPolicyRichtlinien AmazonSSMManagedInstanceCoreund einer IAM-Rolle hinzu. Fügen Sie Ihrer Instance dann die IAM-Rolle als Instanzprofil hinzu. Anweisungen zum Erstellen und Anhängen eines Instance-Profils finden Sie unter Arbeiten mit IAM-Rollen im Amazon EC2 EC2-Benutzerhandbuch.

> **Note**
>
> Sie müssen Amazon Inspector Deep Inspection nicht aktivieren, bevor Sie einen CIS-Scan auf Ihrer Amazon EC2 EC2-Instance ausführen. Wenn Sie Amazon Inspector Deep Inspection deaktivieren, installiert Amazon Inspector automatisch den SSM-Agenten, aber der

SSM-Agent wird nicht mehr aufgerufen, um Deep Inspection auszuführen. Infolgedessen ist die `InspectorLinuxDistributor-do-not-delete` Zuordnung jedoch in Ihrem Konto vorhanden.

Amazon Virtual Private Cloud Cloud-Endpunktanforderungen für die Ausführung von CIS-Scans auf privaten Amazon EC2 EC2-Instances

Sie können CIS-Scans auf Amazon EC2 EC2-Instances über ein Amazon-Netzwerk ausführen. Wenn Sie jedoch CIS-Scans auf privaten Amazon EC2 EC2-Instances ausführen möchten, müssen Sie Amazon VPC-Endpoints erstellen. Die folgenden Endpunkte sind erforderlich, wenn Sie Amazon VPC-Endpoints für Systems Manager erstellen:

- `com.amazonaws.region.inspector2`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ssm`
- `com.amazonaws.region.ssmmessages`

Weitere Informationen finden Sie unter Erstellen von Amazon VPC-Endpunkten für Systems Manager im AWS Systems Manager Benutzerhandbuch.

Note

Ab SSM Agent Version 3.3.40.0 verwendet SSM Agent den `ssmmessages` Endpunkt anstelle des Endpunkts, wann immer dieser verfügbar ist. `ec2messages` Frühere Versionen von SSM Agent verwenden den Endpunkt. `ec2messages` Dieser Endpunkt wird nur unterstützt, wenn er vor 2024 AWS-Regionen gestartet wurde. Weitere Informationen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen im Benutzerhandbuch.AWS Systems Manager

Note

Derzeit unterstützen einige den AWS-Regionen Endpunkt nicht. `amazonaws.com.region.inspector2`

CIS-Scans ausführen

Sie können einen CIS-Scan entweder einmal auf Anforderung oder als geplanten wiederkehrenden Scan ausführen. Um einen Scan auszuführen, müssen Sie zunächst eine Scankonfiguration erstellen.

Wenn Sie eine Scankonfiguration erstellen, geben Sie Tag-Schlüssel-Wert-Paare an, die für Ziel-Instances verwendet werden sollen. Wenn Sie der von Amazon Inspector delegierte Administrator für eine Organisation sind, können Sie in der Scan-Konfiguration mehrere Konten angeben, und Amazon Inspector sucht in jedem dieser Konten nach Instances mit den angegebenen Tags. Sie wählen das CIS-Benchmark-Level für den Scan. Für jeden Benchmark unterstützt CIS ein Level-1- und Level-2-Profil, das Basiswerte für verschiedene Sicherheitsstufen bietet, die verschiedene Umgebungen möglicherweise erfordern.

- Stufe 1 — empfiehlt grundlegende Sicherheitseinstellungen, die auf jedem System konfiguriert werden können. Die Implementierung dieser Einstellungen sollte kaum oder gar nicht zu Betriebsunterbrechungen führen. Ziel dieser Empfehlungen ist es, die Anzahl der Eintrittspunkte in Ihre Systeme zu verringern und so Ihre allgemeinen Cybersicherheitsrisiken zu verringern.
- Stufe 2 — empfiehlt erweiterte Sicherheitseinstellungen für Hochsicherheitsumgebungen. Die Implementierung dieser Einstellungen erfordert Planung und Koordination, um das Risiko geschäftlicher Auswirkungen zu minimieren. Ziel dieser Empfehlungen ist es, Sie bei der Einhaltung gesetzlicher Vorschriften zu unterstützen.

Stufe 2 erweitert Ebene 1. Wenn Sie Level 2 wählen, sucht Amazon Inspector nach allen Konfigurationen, die für Level 1 und Level 2 empfohlen werden.

Nachdem Sie die Parameter für Ihren Scan definiert haben, können Sie wählen, ob Sie ihn als einmaligen Scan, der nach Abschluss der Konfiguration ausgeführt wird, oder als wiederkehrender Scan ausführen möchten. Wiederkehrende Scans können täglich, wöchentlich oder monatlich zu einem Zeitpunkt Ihrer Wahl ausgeführt werden.

> **Tip**
>
> Wir empfehlen, einen Tag und eine Uhrzeit zu wählen, die sich während der Ausführung des Scans am wenigsten auf Ihr System auswirken.

Überlegungen zur Verwaltung von Amazon Inspector CIS-Scans mit AWS Organizations

Wenn Sie CIS-Scans in einer Organisation ausführen, interagieren die von Amazon Inspector delegierten Administratoren und Mitgliedskonten unterschiedlich mit den CIS-Scankonfigurationen und Scanergebnissen.

So können delegierte Administratoren von Amazon Inspector mit CIS-Scankonfigurationen und Scanergebnissen interagieren

Wenn der delegierte Administrator eine Scan-Konfiguration erstellt, entweder für alle Konten oder für ein bestimmtes Mitgliedskonten, ist die Organisation für die Konfiguration verantwortlich. Scankonfigurationen, die einer Organisation gehören, haben einen ARN, der die Organisations-ID als Eigentümer angibt:

`arn:aws:inspector2:`*`Region`*`:`*`111122223333`*`:owner/`*`OrganizationId`*`/cis-configuration/`*`scanId`*

Der delegierte Administrator kann Scankonfigurationen verwalten, die einer Organisation gehören, auch wenn sie von einem anderen Konto erstellt wurden.

Der delegierte Administrator kann die Scanergebnisse für jedes Konto in seiner Organisation einsehen.

Wenn der delegierte Administrator eine Scankonfiguration erstellt und `SELF` als Zielkonto angibt, ist der delegierte Administrator für die Scankonfiguration verantwortlich, auch wenn er das Unternehmen verlässt. Der delegierte Administrator kann das Ziel einer Scankonfiguration mit `SELF` dem Ziel jedoch nicht ändern.

> **Note**
>
> Der delegierte Administrator kann den CIS-Scankonfigurationen, die der Organisation gehören, keine Tags hinzufügen.

So können Amazon Inspector Inspector-Mitgliedskonten mit CIS-Scankonfigurationen und Scanergebnissen interagieren

Wenn ein Mitgliedskonto eine CIS-Scan-Konfiguration erstellt, ist es für die Konfiguration verantwortlich. Der delegierte Administrator kann die Konfiguration jedoch einsehen. Wenn ein

Mitgliedskonto die Organisation verlässt, kann der delegierte Administrator die Konfiguration nicht einsehen.

> **Note**
>
> Der delegierte Administrator kann eine vom Mitgliedskonto erstellte Scan-Konfiguration nicht bearbeiten.

Mitgliedskonten, delegierte Administratoren `SELF` als Ziel und eigenständige Konten erstellen alle ihre eigenen Scankonfigurationen. Diese Scankonfigurationen haben einen ARN, der die Konto-ID als Besitzer anzeigt:

`arn:aws:inspector2:`*`Region`*`:`*`111122223333`*`:owner/`*`111122223333`*`/cis-configuration/`*`scanId`*

Ein Mitgliedskonto kann Scanergebnisse in seinem Konto einsehen, einschließlich der Scanergebnisse von CIS-Scans, die der delegierte Administrator geplant hat.

Amazon Inspector-eigene Amazon S3 S3-Buckets, die für Amazon Inspector CIS-Scans verwendet werden

Open Vulnerability and Assessment Language (OVAL) ist ein Projekt zur Informationssicherheit, das standardisiert, wie der Maschinenzustand von Computersystemen bewertet und gemeldet wird. In der folgenden Tabelle sind alle Amazon S3-Buckets mit OVAL-Definitionen aufgeführt, die Amazon Inspector gehören und für CIS-Scans verwendet werden. Amazon Inspector stellt OVAL-Definitionsdateien bereit, die für CIS-Scans erforderlich sind. Die Amazon S3-Buckets, die Amazon Inspector gehören, sollten bei Bedarf in VPCs auf die Zulassungsliste gesetzt werden.

> **Note**
>
> Die Details für jeden der folgenden Amazon S3-Buckets, die Amazon Inspector gehören, können sich nicht ändern. Die Tabelle kann jedoch aktualisiert werden, um die neu unterstützten AWS-Regionen Funktionen widerzuspiegeln. Sie können Amazon S3-Buckets, die Amazon Inspector gehören, nicht für andere Amazon S3 S3-Operationen oder in Ihren eigenen Amazon S3 S3-Buckets verwenden.

CIS-Eimer	AWS-Region
cis-datasets-prod-arn-5908f6f	Europe (Stockholm)
cis-datasets-prod-bah-8f88801	Middle East (Bahrain)
cis-datasets-prod-bjs-0f40506	China (Peking)
cis-datasets-prod-bom-435a167	Asien-Pazifik (Mumbai)
cis-datasets-prod-cdg-f3a9c58	Europa (Paris)
cis-datasets-prod-cgk-09eb12f	Asien-Pazifik (Jakarta)
cis-datasets-prod-cmh-63030b9	USA Ost (Ohio)
cis-datasets-prod-cpt-02c5c6f	Afrika (Kapstadt)
cis-datasets-prod-dub-984936f	Europa (Irland)
cis-datasets-prod-fra-6eb96eb	Europa (Frankfurt)
cis-datasets-prod-gru-de69f99	Südamerika (São Paulo)
cis-datasets-prod-hkg-8e30800	Asien-Pazifik (Hongkong)
cis-datasets-prod-iad-8438411	USA Ost (Nord-Virginia)
cis-datasets-prod-icn-f4eff1c	Asien-Pazifik (Seoul)
cis-datasets-prod-kix-5743b21	Asien-Pazifik (Osaka)
cis-datasets-prod-lhr-8b1fbd0	Europa (London)
cis-datasets-prod-mxp-7b1bbce	Europa (Milan)
cis-datasets-prod-nrt-464f684	Asien-Pazifik (Tokio)
cis-datasets-prod-osu-5bead6f	AWS GovCloud (US-East)
cis-datasets-prod-pdt-adadf9c	AWS GovCloud (US-West)

CIS-Eimer	AWS-Region
`cis-datasets-prod-pdx-acfb052`	USA West (Oregon)
`cis-datasets-prod-sfo-1515ba8`	USA West (Nordkalifornien)
`cis-datasets-prod-sin-309725b`	Asien-Pazifik (Singapur)
`cis-datasets-prod-syd-f349107`	Asien-Pazifik (Sydney)
`cis-datasets-prod-yul-5e0c95e`	Kanada (Zentral)
`cis-datasets-prod-zhy-5a8eacb`	China (Ningxia)
`cis-datasets-prod-zrh-67e0e3d`	Europa (Zürich)

Eine CIS-Scankonfiguration erstellen

In diesem Thema wird beschrieben, wie eine CIS-Scankonfiguration erstellt wird.

Um einen CIS-Scan auszuführen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie in der AWS-Region Dropdownliste den Ort aus, AWS-Region an dem Sie einen CIS-Scan ausführen möchten.
3. Wählen Sie im Navigationsbereich die Option On-Demand-Scans und dann CIS-Scans aus.
4. Wählen Sie Neuen Scan erstellen aus.
5. Geben Sie unter Name der Scan-Konfiguration einen Namen für die Scan-Konfiguration ein.
6. Geben Sie für Target Resource Tags einen Schlüssel und den entsprechenden Wert für die Instances ein, die Sie scannen möchten. Sie können bis zu fünf verschiedene Werte für jeden Schlüssel und insgesamt 25 Tags angeben, die in den Scan aufgenommen werden sollen.
7. Für das CIS-Benchmark-Level können Sie Level 1 für grundlegende Sicherheitskonfigurationen oder Level 2 für erweiterte Sicherheitskonfigurationen wählen.
8. Geben Sie für Zielkonten an, welche Konten in den CIS-Scan aufgenommen werden sollen. Weitere Informationen finden Sie unter Überlegungen zur Verwaltung von Amazon Inspector CIS-Scans mit AWS Organizations.

Wenn es sich bei Ihrem Konto um das delegierte Administratorkonto handelt, können Sie Alle Konten oder Konten angeben auswählen. Die Option Alle Konten zielt auf alle Konten in Ihrer Organisation ab. Die Option Konten angeben zielt nur auf einzelne Konten in Ihrer Organisation ab. Wenn Sie diese Option wählen, können Sie mehr als ein Konto angeben, indem Sie die Kontonummern durch ein Komma trennen. Sie können `SELF` anstelle einer Konto-ID auch eine Konto-ID eingeben, um eine Scan-Konfiguration für Ihr Konto zu erstellen

Wenn es sich bei Ihrem Konto um ein eigenständiges Konto oder ein Mitgliedskonto in einer Organisation handelt, können Sie Self auswählen, um eine Scankonfiguration für Ihr Konto zu erstellen.

9. Wählen Sie unter Zeitplan die Option Einmaliger Scan, der ausgeführt wird, sobald Sie die Erstellung Ihrer Scankonfiguration abgeschlossen haben, oder Wiederkehrende Scans, der zu dem von Ihnen angegebenen Zeitpunkt ausgeführt wird.
10. Bestätigen Sie Ihre Auswahl und wählen Sie dann Erstellen.

CIS-Scanergebnisse anzeigen

Amazon Inspector erstellt einen Scanauftrag für jede Scankonfiguration, die ausgeführt wird, und sammelt Ergebnisse eines Scans mit einer eindeutigen Scan-ID. Die CIS-Scanergebnisse sind 90 Tage lang verfügbar. Sie können die CIS-Scanergebnisse anhand ihrer Checks oder gescannten Ressourcen anzeigen:

- Nach Prüfungen aggregierte Scanergebnisse — Gruppiert die Ergebnisse eines Scans nach jeder einzelnen Prüfung, die während des Scans durchgeführt wurde. Für jede Prüfung erhalten Sie einen Bericht darüber, wie viele Ressourcen ausgefallen, übersprungen oder bestanden wurden.
- Nach gescannten Ressourcen aggregierte Suchergebnisse — Gruppiert die Ergebnisse eines Scans nach jeder gescannten Ressource, auf die der Scan während des Scans abzielt. Für jede Ressource erhalten Sie einen Bericht darüber, wie viele Prüfungen eine Ressource nicht bestanden, übersprungen oder bestanden hat.

In diesem Thema wird beschrieben, wie die Ergebnisse eines CIS-Scans angezeigt werden.

So zeigen Sie Scanergebnisse an

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.

2. Wählen Sie in der AWS-Region Dropdownliste den Ort aus, an AWS-Region dem Sie Ihre CIS-Scan-Konfiguration erstellt haben.
3. Wählen Sie im Navigationsbereich die Option On-Demand-Scans und dann CIS-Scans aus.
4. Wählen Sie die Registerkarte Scanergebnisse aus.
5. Wählen Sie in der Spalte Geplant nach die ID des Scanzeitplans aus, den Sie anzeigen möchten. Oder wählen Sie die Zeile mit der Scanzeitplan-ID aus, die Sie anzeigen möchten, und wählen Sie dann Details anzeigen.
6. Wählen Sie „Prüfungen“, um alle durchgeführten Prüfungen anzuzeigen, oder „Gescannte Ressourcen“, um alle gescannten Ressourcen anzuzeigen, die während des Scans als Ziel ausgewählt wurden.

Sie können auch Details für geplante CIS-Scans anzeigen.

Um Details für geplante CIS-Scans anzuzeigen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie in der AWS-Region Dropdownliste den Ort aus, an AWS-Region dem Sie Ihre CIS-Scan-Konfiguration erstellt haben.
3. Wählen Sie im Navigationsbereich die Option On-Demand-Scans und dann CIS-Scans aus.
4. Wählen Sie die Registerkarte Geplant.
5. Wählen Sie in der Spalte Name der Scankonfiguration den Namen der Scankonfiguration aus, die Sie anzeigen möchten. Oder wählen Sie die Zeile mit der Scankonfiguration aus, die Sie anzeigen möchten, und wählen Sie dann Details anzeigen.

Bearbeitung einer CIS-Scankonfiguration

In diesem Thema wird beschrieben, wie eine CIS-Scankonfiguration bearbeitet wird.

Um eine CIS-Scankonfiguration zu bearbeiten

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie in der AWS-Region Dropdownliste den Ort aus, an AWS-Region dem Sie Ihre CIS-Scan-Konfiguration erstellt haben.

3. Wählen Sie im Navigationsbereich die Option On-Demand-Scans und dann CIS-Scans aus.
4. Wählen Sie die Registerkarte Geplant.
5. Wählen Sie die Zeile mit der Scankonfiguration aus, die Sie bearbeiten möchten, und klicken Sie dann auf Bearbeiten.

CIS-Scanergebnisse herunterladen

Sie können eine PDF- oder CSV-Datei eines CIS-Scans mithilfe der Amazon Inspector Inspector-Konsole oder API herunterladen.

> **Note**
>
> Sie können nur eine CSV-Datei mit Ihren CIS-Scanergebnissen für CIS-Scans herunterladen, die nach dem 05.03.2024 erfasst wurden.

In diesem Thema wird beschrieben, wie Sie einen CIS-Scan mithilfe der Amazon Inspector Inspector-Konsole herunterladen.

Um CIS-Scanergebnisse von der Konsole herunterzuladen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie in der AWS-Region Dropdownliste den Ort aus, an AWS-Region dem Sie Ihre CIS-Scan-Konfiguration erstellt haben.
3. Wählen Sie im Navigationsbereich die Option On-Demand-Scans und dann CIS-Scans aus.
4. Wählen Sie die Registerkarte Scanergebnisse aus.
5. Wählen Sie in der Spalte Geplant von die ID des Scanzeitplans aus, die Sie anzeigen möchten. Oder wählen Sie die Zeile mit der Scanzeitplan-ID aus, die Sie anzeigen möchten, und wählen Sie dann Details anzeigen.
6. Wählen Sie „Herunterladen“ und dann „PDF“ oder „CSV“. Wenn es sich bei Ihrem Konto um das delegierte Administratorkonto handelt, können Sie Konto auswählen wählen, um Ergebnisse für ein bestimmtes Mitgliedskonto herunterzuladen.

Amazon Inspector Codesicherheit

Amazon Inspector ist ein Schwachstellen-Management-Service, der Workloads automatisch erkennt und sie kontinuierlich auf Softwareschwachstellen und unbeabsichtigte Netzwerkbedrohungen überprüft. Mit Code Security scannt Amazon Inspector den Quellcode von Erstanbieteranwendungen, Abhängigkeiten von Drittanbieteranwendungen und Infrastructure as Code auf Sicherheitslücken. Sie können Code Security in der Amazon Inspector Inspector-Konsole oder mit der Amazon Inspector Inspector-API aktivieren. Sobald Sie Code Security aktiviert haben, können Sie eine Scan-Konfiguration erstellen und auf Ihr Code-Repository anwenden, um festzulegen, wie oft und wann der Code gescannt wird. Sie können Ihre Scankonfiguration jederzeit anzeigen, bearbeiten und löschen. Informationen darüber, AWS-Regionen wo Code Security verfügbar ist, finden Sie unter Regionen und Endpunkte. Preisinformationen finden Sie unter Amazon Inspector — Preise.

Voraussetzungen für die Codesicherheit

Bevor Sie Code Security verwenden können, müssen Sie Code Security aktivieren und entscheiden, wie Sie Ihre Daten verschlüsseln möchten. Dies können Informationen wie Anmeldeinformationen für die Integration, Code oder andere Informationen sein, die sich auf Ihre Integrationen, Code-Repositorys und Projekte beziehen. Standardmäßig werden Ihre Daten mit einem AWS eigenen Schlüssel verschlüsselt. Das bedeutet, dass der Schlüssel vom Dienst erstellt wird, ihm gehört und von ihm verwaltet wird. Wenn Sie den Schlüssel, der zur Verschlüsselung Ihrer Daten verwendet wird, besitzen und verwalten möchten, können Sie einen vom Kunden verwalteten KMS-Schlüssel erstellen.

Code-Sicherheit aktivieren

Sie aktivieren Code Security auf die gleiche Weise wie alle automatisierten Scantypen. Weitere Informationen finden Sie unter Einen Suchtyp aktivieren.

Einen vom Kunden verwalteten Zugriffsschlüssel erstellen AWS KMS

Standardmäßig werden Ihre Daten mit einem AWS eigenen Schlüssel verschlüsselt. Das bedeutet, dass der Schlüssel vom Dienst erstellt wird, ihm gehört und von ihm verwaltet wird. Wenn Sie den Schlüssel, der zur Verschlüsselung Ihrer Daten verwendet wird, besitzen und verwalten möchten, können Sie einen vom Kunden verwalteten KMS-Schlüssel erstellen. Amazon Inspector interagiert nicht mit Ihren Daten. Amazon Inspector nimmt nur Metadaten aus Repositorys in Ihrem Quellcode-

Anbieter auf. Informationen zum Erstellen eines vom Kunden verwalteten KMS-Schlüssels finden Sie unter Erstellen eines KMS-Schlüssels im AWS Key Management Service Benutzerhandbuch.

Beispiel für eine Richtlinie

Verwenden Sie beim Erstellen Ihres vom Kunden verwalteten Schlüssels die folgende Beispielrichtlinie.

> **Note**
>
> Die FAS-Berechtigungen in der folgenden Richtlinie sind spezifisch für Amazon Inspector, da sie es Amazon Inspector ermöglichen, nur diese API-Aufrufe durchzuführen.

JSON

```
{
  "Version":"2012-10-17",
  "Id": "key-policy",
  "Statement": [
    {
      "Sid": "Allow Q to use Encrypt Decrypt GenerateDataKey and
 GenerateDataKeyWithoutPlaintext",
      "Effect": "Allow",
      "Principal": {
        "Service": "q.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
        "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
        "kms:EncryptionContext:aws:qdeveloper:codesecurity-scope": "111122223333"
        },
        "ArnLike": {
```

```
        "aws:SourceArn": "arn:aws:inspector2:us-east-1:111122223333:codesecurity-
integration/*"
        }
      }
    },
    {
      "Sid": "Allow Q to use DescribeKey",
      "Effect": "Allow",
      "Principal": {
        "Service": "q.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "Allow Inspector to use Encrypt Decrypt GenerateDataKey and
 GenerateDataKeyWithoutPlaintext using FAS",
      "Effect": "Allow",
      "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/inspectorCodeSecurity"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "inspector2.us-east-1.amazonaws.com"
        },
        "StringLike": {
        "kms:EncryptionContext:aws:qdeveloper:codesecurity-scope": "111122223333"
        }
      }
    },
    {
      "Sid": "Allow Inspector to use DescribeKey using FAS",
      "Effect": "Allow",
      "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/inspectorCodeSecurity"
      },
      "Action": [
```

```
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "inspector2.us-east-1.amazonaws.com"
        }
      }
    }
  ]
}
```

Nachdem Sie Ihren KMS-Schlüssel erstellt haben, können Sie den folgenden Amazon Inspector verwenden APIs.

- UpdateEncryptionKey — Verwenden Sie mit `CODE_REPOSITORY` for `resourceType` und `CODE` als Scan-Typ, um die Verwendung Ihres vom Kunden verwalteten KMS-Schlüssels zu konfigurieren.
- GetEncryptionKey — Verwenden Sie mit `CODE_REPOSITORY` for `resourceType` und `CODE` als Suchtyp, um den Abruf Ihrer KMS-Schlüsselkonfiguration zu konfigurieren.
- ResetEncryptionKey — Verwenden Sie mit `CODE_REPOSITORY` for `resourceType` und`CODE`, um Ihre KMS-Schlüsselkonfiguration zurückzusetzen und einen AWS eigenen KMS-Schlüssel zu verwenden.

Eine Integration zwischen Amazon Inspector und Ihrem Code-Repository erstellen

Dieser Abschnitt enthält Themen, in denen beschrieben wird, wie Sie eine Integration zwischen Amazon Inspector und Ihrem Code-Repository erstellen. Wenn Sie eine Integration erstellen, werden alle Code-Repositorys in der Amazon Inspector Inspector-Konsole auf der Seite Code Security als Projekte aufgeführt. In anderen Themen in diesem Abschnitt wird beschrieben, wie Sie auf Ihre Integrationen und Projekte zugreifen können.

Code Security importiert nur bis zu 100.000 Projekte, und nur der Standardzweig für jedes Repository wird überwacht. Ein Projekt kann maximal drei Standard-Scankonfigurationen zugeordnet werden.

Code Security unterstützt nur maximal 100 Integrationen pro Konto. Bei den Integrationen von Code Security gibt es kein Konzept für die Beziehung zwischen delegierten account/member Administratorkonten.

Um Einschränkungen zu vermeiden, empfehlen wir, denselben Host nicht mehrmals für eine Integration zu verwenden.

Integrationen mit GitHub SaaSGitHub Enterprise Cloud, und GitHub Enterprise Server erfordern einen öffentlichen Internetzugang.

> **Important**
>
> Integrationen von Drittanbietern können ohne vorherige Ankündigung aus beliebigem Grund vorübergehend oder dauerhaft deaktiviert werden, z. B. um Sicherheitsbedenken auszuräumen.

Erstellen einer Integration zwischen Amazon Inspector und GitHub

In diesem Thema wird beschrieben, wie Sie eine Integration zwischen Amazon Inspector und erstellenGitHub.

> **Note**
>
> Wenn Sie zum ersten Mal eine Integration erstellen, werden Sie in Schritt 2 aufgefordert, eine Standard-Scan-Konfiguration zu erstellen. Wenn Sie eine Scankonfiguration erstellen, wählen Sie die Scanfrequenz, die Scananalyse und die zu scannenden Repositorys aus. Das Erstellen einer Standard-Scankonfiguration entspricht dem Erstellen einer allgemeinen Scankonfiguration. Die Standard-Scankonfiguration wird jedoch automatisch allen neuen und vorhandenen Projekten zugeordnet, die in Amazon Inspector importiert werden. Wenn Sie eine Standard-Scan-Konfiguration erstellen möchten, wählen Sie Mit dieser Konfiguration fortfahren. Sie können eine Standard-Scan-Konfiguration nur einmal erstellen. Wenn Sie eine Standard-Scankonfiguration erstellen, werden Sie nicht erneut aufgefordert, eine Standard-Scankonfiguration zu erstellen. Sie können eine Standard-Scankonfiguration nur einmal pro Konto und einmal pro Organisation erstellen. Wenn Sie keine Standard-Scan-Konfiguration konfigurieren möchten, wählen Sie Konfiguration überspringen. Sie werden jedoch aufgefordert, eine Standard-Scankonfiguration zu erstellen, wenn Sie das nächste Mal eine Integration erstellen. Nachdem Sie eine Standard-Scan-Konfiguration erstellt oder die

Erstellung einer Standard-Scan-Konfiguration übersprungen haben, werden Sie zu Schritt 3 des Integrations-Workflows weitergeleitet, wo Sie Ihre Integrationsdetails eingeben.

Integrationen mit GitHub SaaSGitHub Enterprise Cloud, und GitHub Enterprise Server erfordern einen öffentlichen Internetzugang.

> **Note**
>
> Amazon Inspector scannt und überwacht nur Ihre Standardfiliale. Wenn Sie einen neuen Standardzweig erstellen, scannt und aktualisiert Amazon Inspector den neuen Standardzweig.

> **Important**
>
> Bevor Sie mit der Erstellung der Integration fertig sind, werden Sie angewiesen, die Verbindung zwischen Amazon Inspector und GitHub zu autorisieren. Sie müssen diesen Schritt abschließen, um den Vorgang abzuschließen. Wenn Sie das Pop-up schließen, können Sie nicht fortfahren.

Um eine Integration zwischen Amazon Inspector und GitHub

1. Melden Sie sich mit Ihren Zugangsdaten an. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus. Wählen Sie Connect und wählen SieGitHub.
3. Geben Sie unter Integrationsdetails den Namen Ihrer Integration ein und wählen Sie Connect aus GitHub.
4. Wählen Sie im Pop-up Autorisieren, um eine Verbindung zwischen Amazon Inspector und GitHub herzustellen.
5. Wählen Sie im Erfolgsbanner die Option Gehe zur Seite zur GitHub Verbindungserstellung.
6. Geben Sie die Installations-ID für die GitHub Anwendung ein. Wenn Sie die GitHub Anwendung installiert haben, finden Sie die Installations-ID auf GitHub der GitHub Apps-Seite oder am Ende der GitHub Anwendungs-URL. Wenn Sie die GitHub Anwendung nicht installiert haben, wählen

Sie Neue App installieren. Dadurch werden Sie zu GitHub dem Punkt weitergeleitet, an dem Sie die GitHub Organisation auswählen und den Repository-Bereich angeben.

7. Wählen Sie Connect GitHub.

Nachdem Sie die Integration erstellt haben, kann ein Szenario auftreten, in dem Amazon Inspector das Zugriffstoken nicht aktualisieren kann. Dies kann der Fall sein, wenn der Integrationshost nicht verfügbar ist oder Amazon Inspector andere Kommunikationsprobleme hat. Um das Problem zu beheben, können Sie die Verbindung über den Tab Integrationen auf der Seite Codesicherheit erneut authentifizieren. In der Spalte Status wird die Integration als Inaktiv angezeigt, und Amazon Inspector bietet die Option zur erneuten Authentifizierung. Wählen Sie Erneut authentifizieren. Sie werden zum Integrations-Workflow weitergeleitet, wo Sie die Verbindungseinrichtung abschließen können.

Wenn Sie die Systemeinstellungen für Ihre Integration löschen, kann die Verbindung auf unbestimmte Zeit unterbrochen werden. In diesem Fall müssen Sie die Integration löschen und eine neue Integration erstellen. Wenn Sie eine Integration löschen, verlieren Sie alle Projekte und Scankonfigurationen, die mit der Integration verknüpft sind.

Erstellen einer Integration zwischen Amazon Inspector und GitLab Self Managed

In diesem Thema wird beschrieben, wie Sie eine Integration zwischen Amazon Inspector und Ihrem Code-Repository in erstellenGitLab Self Managed.

Erforderliche Informationen

Folgendes ist erforderlich, wenn Sie eine Verbindung herstellen:

- Integrationsname — Dies ist der Name, der dem Hauptteil Ihrer Integration hinzugefügt wurde.
- Endpunkt-URL — Dies ist die URL, die für den Zugriff auf Ihre GitLab Self Managed Instanz verwendet wird.
- Persönliches Zugriffstoken — Das persönliche Zugriffstoken wird GitLab Self Managed von einem Administratorkonto aus erstellt und muss die folgenden Bereiche enthalten: `apiread_api`, `undread_repository`.

> **Note**
>
> Amazon Inspector scannt und überwacht nur Ihre Standardfiliale. Wenn Sie einen neuen Standardzweig erstellen, scannt und aktualisiert Amazon Inspector den neuen Standardzweig.

Erstellen einer Integration zwischen Amazon Inspector und GitLab Self Managed

Das folgende Verfahren beschreibt, wie Sie eine Verbindung zwischen Amazon Inspector und Ihrem Code-Repository in herstellenGitLab Self Managed.

> **Note**
>
> Wenn Sie zum ersten Mal eine Integration erstellen, werden Sie in Schritt 2 aufgefordert, eine Standard-Scan-Konfiguration zu erstellen. Wenn Sie eine Scankonfiguration erstellen, wählen Sie die Scanfrequenz, die Scananalyse und die zu scannenden Repositorys aus. Das Erstellen einer Standard-Scankonfiguration entspricht dem Erstellen einer allgemeinen Scankonfiguration. Die Standard-Scankonfiguration wird jedoch automatisch allen neuen und vorhandenen Projekten zugeordnet, die in Amazon Inspector importiert werden. Wenn Sie eine Standard-Scan-Konfiguration erstellen möchten, wählen Sie Mit dieser Konfiguration fortfahren. Sie können eine Standard-Scan-Konfiguration nur einmal erstellen. Wenn Sie eine Standard-Scankonfiguration erstellen, werden Sie nicht erneut aufgefordert, eine Standard-Scankonfiguration zu erstellen. Sie können eine Standard-Scankonfiguration nur einmal pro Konto und einmal pro Organisation erstellen. Wenn Sie keine Standard-Scan-Konfiguration konfigurieren möchten, wählen Sie Konfiguration überspringen. Wenn Sie das nächste Mal eine Integration erstellen, werden Sie jedoch aufgefordert, eine Standard-Scankonfiguration zu erstellen. Nachdem Sie eine Standard-Scan-Konfiguration erstellt oder die Erstellung einer Standard-Scan-Konfiguration übersprungen haben, werden Sie zu Schritt 3 des Integrations-Workflows weitergeleitet, wo Sie Ihre Integrationsdetails eingeben.

> **Important**
>
> Bevor Sie mit der Erstellung der Integration fertig sind, werden Sie aufgefordert, die Verbindung zwischen Amazon Inspector und GitLab Self Managed zu autorisieren. Sie

müssen diesen Schritt abschließen, um den Vorgang abzuschließen. Wenn Sie das Pop-up schließen, können Sie nicht fortfahren.

Um eine Verbindung mit GitLab Self Managed herzustellen

1. Melden Sie sich mit Ihren Zugangsdaten an. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus. Wählen Sie Connect und dann GitLab Self Managed aus.
3. Geben Sie unter Integrationsdetails Folgendes ein:
    a. Geben Sie unter Integrationsname den Namen ein, der dem Hauptteil Ihrer Integration hinzugefügt wurde.
    b. Geben Sie unter Endpunkt-URL die URL ein, die für den Zugriff auf Ihre GitLab selbstverwaltete Instanz verwendet wird.
    c. Geben Sie unter Persönliches Zugriffstoken Ihr persönliches Zugriffstoken mit den erforderlichen Gültigkeitsbereichen ein.
4. Wählen Sie Verbinden mit. GitLab
5. Wählen Sie im Popup-Fenster die Option Autorisieren, um die Herstellung einer Verbindung zwischen Amazon Inspector und GitLab abzuschließen.

Nachdem Sie die Integration erstellt haben, kann ein Szenario auftreten, in dem Amazon Inspector das Zugriffstoken nicht aktualisieren kann. Dies kann der Fall sein, wenn der Integrationshost nicht verfügbar ist oder Amazon Inspector andere Kommunikationsprobleme hat. Um das Problem zu beheben, können Sie die Verbindung über den Tab Integrationen auf der Seite Code Security erneut authentifizieren. In der Spalte Status wird die Integration als Inaktiv angezeigt, und Amazon Inspector bietet die Option zur erneuten Authentifizierung. Wählen Sie Erneut authentifizieren. Sie werden zum Integrations-Workflow weitergeleitet, wo Sie die Verbindungseinrichtung abschließen können.

Wenn Sie die Systemeinstellungen für Ihre Integration löschen, kann die Verbindung auf unbestimmte Zeit unterbrochen werden. In diesem Fall müssen Sie die Integration löschen und eine neue Integration erstellen. Wenn Sie eine Integration löschen, verlieren Sie alle Projekte und Scankonfigurationen, die mit der Integration verknüpft sind.

Integrationen mit Code-Repositorys anzeigen

In diesem Thema wird beschrieben, wie Integrationen in der Amazon Inspector Inspector-Konsole angezeigt werden.

So zeigen Sie Integrationen in der Amazon Inspector Inspector-Konsole an

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wähle Sie Integrations (Integrationen) aus. Auf dieser Registerkarte können Sie alle Ihre konfigurierten Integrationen und grundlegende Informationen zu all Ihren Integrationen überprüfen. Zu diesen Informationen gehören der Name der Integration, der Status der Integration und der Name des Quellcode-Anbieters.

Authentifizieren Sie sich erneut beim Anbieter

Nachdem Sie die Integration erstellt haben, kann ein Szenario auftreten, in dem Amazon Inspector das Zugriffstoken nicht aktualisieren kann. Dies kann der Fall sein, wenn der Integrationshost nicht verfügbar ist oder Amazon Inspector andere Kommunikationsprobleme hat. Um das Problem zu beheben, können Sie die Verbindung über den Tab Integrationen auf der Seite Codesicherheit erneut authentifizieren. In der Spalte Status wird die Integration als Inaktiv angezeigt, und Amazon Inspector bietet die Option zur erneuten Authentifizierung. Wählen Sie Erneut authentifizieren. Sie werden zum Integrations-Workflow weitergeleitet, wo Sie die Verbindungseinrichtung abschließen können.

Wenn Sie die Systemeinstellungen für Ihre Integration löschen, kann die Verbindung auf unbestimmte Zeit unterbrochen werden. In diesem Fall müssen Sie die Integration löschen und eine neue Integration erstellen. Wenn Sie eine Integration löschen, verlieren Sie alle Projekte und Scankonfigurationen, die mit der Integration verknüpft sind.

Code-Repositorys anzeigen

In diesem Thema wird beschrieben, wie Code-Repositorys in der Amazon Inspector Inspector-Konsole angezeigt werden.

Um Code-Repositorys in der Amazon Inspector Inspector-Konsole anzuzeigen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.

2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wählen Sie Code-Repositorys aus. Auf dieser Registerkarte können Sie alle Ihre Code-Repositorys, die als Projekte aufgeführt sind, sowie grundlegende Informationen zu ihnen überprüfen. Zu diesen Informationen gehören der Name und der Scanstatus für jedes Projekt. Sie können auch die Konfigurationen überprüfen, die Ihren Projekten zugeordnet sind und wann Ihre Projekte zuletzt gescannt wurden. Sie können Ihre Projekte sogar in der Suchleiste filtern.

Details für ein Projekt anzeigen

In diesem Thema wird beschrieben, wie Sie Details für ein Projekt in der Amazon Inspector Inspector-Konsole anzeigen. Wenn Ihr Konto der delegierte Administrator für eine Organisation ist, können Sie Details zu Projekten anzeigen, die zu Mitgliedskonten gehören.

So zeigen Sie Codeprojekte in der Amazon Inspector Inspector-Konsole an

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wählen Sie Code-Repositorys aus. Auf dieser Registerkarte können Sie alle Ihre Code-Repositorys, die als Projekte aufgeführt sind, sowie grundlegende Informationen zu ihnen überprüfen. Zu diesen Informationen gehören der Name und der Scanstatus für jedes Projekt. Sie können auch die mit Ihren Projekten verknüpften Konfigurationen und den Zeitpunkt überprüfen, an dem Ihre Projekte zuletzt gescannt wurden. Sie können Ihre Projekte sogar in der Suchleiste filtern.
4. Wählen Sie ein Projekt aus. Oder wählen Sie ein Projekt aus und wählen Sie Details anzeigen. Auf dem Bildschirm mit den Projektdetails können Sie grundlegende Informationen über das Projekt überprüfen. Zu diesen Informationen gehören der Name und die ID für das Projekt sowie der Integrations-ARN. Sie enthalten Informationen darüber, wann das Projekt gescannt wurde und um welchen Anbietertyp es sich handelt. Sie können sogar die mit dem Projekt verknüpften Ergebnisse überprüfen, Ergebnisse exportieren und Regeln zur Unterdrückung von Ergebnissen erstellen.

Eine Integration löschen

Das folgende Verfahren beschreibt, wie Sie eine Integration in der Amazon Inspector Inspector-Konsole löschen. Wenn Sie eine Integration löschen, verlieren Sie alle Projekte und Scankonfigurationen, die mit der Integration verknüpft sind.

Um eine Integration in der Amazon Inspector Inspector-Konsole zu löschen.

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wähle Sie Integrations (Integrationen) aus. Auf dieser Registerkarte können Sie alle Ihre konfigurierten Integrationen und grundlegende Informationen zu all Ihren Integrationen überprüfen. Zu diesen Informationen gehören der Name der Integration, der Status der Integration und der Typ des Integrationsanbieters.
4. Wählen Sie eine Integration aus und klicken Sie auf Löschen.

Eine Scan-Konfiguration erstellen

Bevor Sie eine Scankonfiguration erstellen, müssen Sie eine Integration mit Amazon Inspector erstellen. Wenn Sie zum ersten Mal eine Integration erstellen, werden Sie aufgefordert, eine Standard-Scan-Konfiguration zu erstellen. In diesem Thema wird beschrieben, wie Sie eine allgemeine Scankonfiguration erstellen. Der Unterschied zwischen einer Standard-Scankonfiguration und einer allgemeinen Scankonfiguration besteht darin, dass neuen Projekten automatisch eine Standardscankonfiguration zugewiesen wird. Sie können die Erstellung einer Standard-Scankonfiguration überspringen.

Code Security unterstützt nur maximal 500 allgemeine Scankonfigurationen. Code Security unterstützt nur eine Standard-Scankonfiguration pro Konto und Organisation. Eine Scankonfiguration kann nur mit maximal 100.000 Projekten verknüpft werden.

Ein Projekt kann insgesamt maximal 4 Scankonfigurationen zugeordnet werden. Dies beinhaltet eine Standard-Scankonfiguration, falls eine Standard-Scankonfiguration erstellt wurde. Scankonfigurationen für eine Organisation können nicht markiert werden.

Wenn der delegierte Administrator für eine Organisation eine Scankonfiguration erstellt, wird die Scankonfiguration auf Organisationsebene erstellt und auf alle Mitgliedskonten in der Organisation

angewendet. Das Gleiche gilt, wenn der delegierte Administrator eine Standardscankonfiguration erstellt.

Wenn Sie eine Scankonfiguration erstellen, wählen Sie die Scanfrequenz, die Scananalyse und die zu scannenden Repositorys aus. Die Scanfrequenz kann je nach Bedarf und in regelmäßigen Abständen oder individuell angepasst werden. Bei änderungsbasiertem und regelmäßigem Scannen haben Sie die Möglichkeit, das regelmäßige Scannen zu aktivieren. Wenn Sie das regelmäßige Scannen aktivieren, legen Sie die Scanfrequenz auf den Wochentag oder den Monat fest, an dem ein Scan durchgeführt wird. Benutzerdefiniertes Scannen bietet Ihnen die Möglichkeit, das Scannen bei Codeänderungen und das regelmäßige Scannen zu aktivieren. Wenn Sie das Scannen bei Codeänderungen aktivieren, geben Sie den Scan-Trigger an, der in Merge- und Pull-Anfragen aufgenommen werden soll.

Scans können übersprungen werden, wenn sich eine Commit-ID innerhalb eines festgelegten Zeitraums nicht geändert hat. Bei regelmäßigen Scans werden Scans übersprungen, wenn sich eine Commit-ID zwischen den Scans innerhalb einer Woche nicht geändert hat. Bei On-Demand-Scans werden Scans übersprungen, wenn sich eine Commit-ID zwischen den Scans innerhalb von 24 Stunden nicht geändert hat.

> **Note**
>
> Wenn eine Scankonfiguration nur Trigger für Merge- und Pull-Requests enthält, werden nur die 25 wichtigsten oder wichtigsten Ergebnisse angezeigt, und zwar nur in der Quellcode-Management-Plattform. Keine wird in Amazon Inspector sichtbar sein.

Um eine allgemeine Scan-Konfiguration zu erstellen

1. Melden Sie sich mit Ihren Zugangsdaten an. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wählen Sie Konfigurationen und anschließend Scankonfiguration erstellen aus.
4. Gehen Sie unter Scandetails wie folgt vor:

   - Geben Sie unter Konfigurationsname einen Namen für die Scankonfiguration ein.
5. Geben Sie unter Scan-Frequenz an, wie oft Code gescannt wird, indem Sie auf Änderung basierendes und periodisches Scannen oder Benutzerdefinierte Scantypen und -auslöser auswählen.

a. (Option 1) Wenn Sie auf Änderungen basierendes und periodisches Scannen wählen, wählen Sie Periodische Suche aktivieren oder Periodische Suche deaktivieren aus.

   - . Wenn Sie Periodischen Scan aktivieren wählen, legen Sie die Scanfrequenz fest, indem Sie die Woche und den Tag auswählen, an dem der Code gescannt werden soll.

b. (Option 2) Wenn Sie Benutzerdefiniertes Scannen wählen, entscheiden Sie, ob das Scannen bei Codeänderungen und das regelmäßige Scannen aktiviert werden sollen.

   i. Wählen Sie Scannen bei Codeänderung aktivieren oder Scannen bei Codeänderung deaktivieren. Wenn Sie „Scannen bei Codeänderung aktivieren“ wählen, geben Sie in der Dropdownliste an, wann Scans ausgelöst werden sollen.

   ii. Wählen Sie Periodischen Scan aktivieren oder Periodischen Scan deaktivieren. Wenn Sie Periodischen Scan aktivieren wählen, legen Sie die Scan-Frequenz fest, indem Sie die Woche und den Tag auswählen, an dem der Code gescannt werden soll. Sie können auch nach ereignisbasierten Triggern scannen. Zu diesen Ereignissen gehört, wenn ein neuer Pull-Request zum ersten Mal für den Standard-Branch geöffnet wird und wenn ein Commit zusammengeführt oder in den Standard-Branch übertragen wird. Bei nachfolgenden Aktualisierungen oder Überarbeitungen einer bestehenden Pull-Anfrage werden keine Scans ausgelöst. Um einen neuen Scan auszulösen, schließen Sie den Pull-Request und öffnen Sie ihn erneut.

6. Entscheiden Sie unter Scananalyse, ob Sie eine vollständige Scananalyse oder eine benutzerdefinierte Scananalyse konfigurieren möchten:

   a. (Option 1) Wenn Sie Vollständige Scananalyse wählen, wenden Sie alle der folgenden Scananalysen an:

      - Statische Anwendungssicherheitstests — Analysiert den Quellcode auf Sicherheitslücken.
      - IaC-Scanning — Analysiert Skripts und Code, mit denen die Infrastruktur konfiguriert und bereitgestellt wird.
      - Statische Analyse der Softwarezusammensetzung — Untersucht Open-Source-Pakete in Anwendungen.

   b. (Option 2) Wenn Sie die Option Benutzerdefinierte Scananalyse wählen, müssen Sie mindestens einen Typ der zuvor genannten Scananalysetypen aus dem Drop-down-Menü auswählen:

7. (Optional) Erstellen Sie für Tags ein Schlüssel-Wert-Paar, das auf Ihr Projekt angewendet werden soll. Sie können bis zu 50 Tags erstellen.

8. Wählen Sie Weiter aus.
9. Wählen Sie unter Repository-Auswahl die Option Alle Repositorys oder Spezifische Repositorys aus.
    a. (Option 1) Wenn Sie Alle Repositorys wählen, ist das Scannen für alle vorhandenen Repositorys aktiviert.
    b. (Option 2) Wenn Sie Spezifische Repositorys wählen, ist das Scannen nur für die von Ihnen angegebenen Repositorys aktiviert.
10. Wählen Sie Weiter aus.
11. Überprüfen Sie Ihre Auswahl und wählen Sie dann Scankonfiguration erstellen aus.

> **Note**
>
> Allgemeine Scankonfigurationen werden nur auf alle vorhandenen Code-Repositorys angewendet. Sie werden nicht auf neue Code-Repositorys angewendet.

Scan-Konfigurationen anzeigen

Das folgende Verfahren beschreibt, wie Sie Scankonfigurationen in der Amazon Inspector Inspector-Konsole anzeigen.

> **Note**
>
> Wenn Sie sich Ihre Scan-Konfiguration auf Organisationsebene ansehen, können einige Details auf dem Code-Sicherheitsbildschirm je nach Ihren Anforderungen abweichen AWS-Konto.

Um Details zu einer Scan-Konfiguration anzuzeigen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wählen Sie Konfigurationen, um eine Liste Ihrer Scankonfigurationen anzuzeigen. Wenn Sie der delegierte Administrator sind, enthält die Liste die Scankonfigurationen Ihres Unternehmens.

Sie können den Namen jeder Scan-Konfiguration sehen und sehen, wer die einzelnen Scan-Konfigurationen erstellt hat (AWS-Konto ID oder Organisations-ID). Sie können auch sehen, welche Scantypen und welcher Scananalysetyp auf die Konfiguration angewendet werden. Sie können Ihre Scankonfiguration sogar nach verschiedenen Feldern in der Suchleiste filtern.

Details für eine Scan-Konfiguration anzeigen

Das folgende Verfahren beschreibt, wie Sie Details für eine Scan-Konfiguration in der Amazon Inspector Inspector-Konsole anzeigen.

Um Details zu einer Scan-Konfiguration anzuzeigen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wählen Sie Konfigurationen aus.
4. Wählen Sie die Konfiguration aus, für die Sie Details anzeigen möchten. Der Bildschirm mit den Details zur Scankonfiguration bietet einen Überblick über die Scankonfiguration. Auf diesem Bildschirm können Sie den ARN für die Scankonfiguration einsehen, welche Scanfrequenztypen aktiviert sind und welche Scananalysetypen aktiviert sind. Sie können die Scankonfiguration auch von diesem Bildschirm aus löschen. Wenn Sie sich eine Scankonfiguration ansehen, die zu Ihrer Organisation gehört, können Sie sie auch von diesem Bildschirm aus bearbeiten.

Bearbeitung einer Scan-Konfiguration

Sie können eine Scankonfiguration jederzeit bearbeiten. Wenn Sie eine Scankonfiguration bearbeiten, können Sie die Scanfrequenz, die Scananalyse, die Tags und die zu scannenden Repositorys ändern. Sie bearbeiten beispielsweise eine Scankonfiguration, um das Scannen für ein bestimmtes Repository anzuhalten. Das folgende Verfahren beschreibt, wie Sie eine Scankonfiguration bearbeiten.

Um eine Scankonfiguration zu bearbeiten

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Code Security aus.
3. Wählen Sie Konfigurationen aus.

4. Wählen Sie die Konfiguration aus, die Sie bearbeiten möchten, und klicken Sie dann auf Bearbeiten. Sie können auch die Konfiguration auswählen, die Sie bearbeiten möchten, und dann Bearbeiten wählen.

Löschen einer Scan-Konfiguration

Sie können eine Scankonfiguration jederzeit löschen. In diesem Thema wird beschrieben, wie eine Scankonfiguration gelöscht wird.

Um eine Scan-Konfiguration zu löschen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich die Option Codesicherheit aus.
3. Wählen Sie Konfigurationen aus.
4. Wählen Sie die Konfiguration aus, die Sie löschen möchten, und wählen Sie dann Löschen. Oder wählen Sie die Konfiguration aus, die Sie löschen möchten, und wählen Sie dann Löschen.

Einen Scan auf Anforderung durchführen

Sie können einen On-Demand-Service für Ihre Projekte durchführen. Wenn Sie einen Scan auf Anforderung durchführen, wird eine Vereinigung all Ihrer konfigurierten Scankonfigurationen auf Ihr ausgewähltes Projekt angewendet. Wenn Ihr Konto das delegierte Administratorkonto für eine Organisation ist, können Sie einen On-Demand-Scan für Projekte durchführen, die zu Mitgliedskonten gehören. Das folgende Verfahren beschreibt, wie Sie einen On-Demand-Scan in der Amazon Inspector Inspector-Konsole durchführen.

Um einen On-Demand-Scan durchzuführen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich die Option Code-Sicherheit aus.
3. Wählen Sie Code-Repositorys aus.
4. Wählen Sie das Projekt aus, das Sie scannen möchten, und wählen Sie dann Scannen auf Anforderung.

Unterstützte Sprachen für Amazon Inspector Inspector-Codesicherheit

Dieses Thema umfasst die unterstützten Sprachen für Amazon Inspector Code Security.

Unterstützte Sprachen für SAST

- C#(Alle Versionen außer .Net 6.0 und höher werden empfohlen)
- C(C11 oder früher)
- C++(C++17 oder früher)
- Go(nur Go 1.18)
- Java(Java25 oder früher)
- JavaScript(EMCMAScript 2021 oder früher)
- JSX(React 17 oder früher)
- Kotlin(Kotlin2.0 oder früher)
- PHP(PHP8.2 oder früher)
- Python(Python3.13 oder früher innerhalb der Serie Python 3)
- Ruby(nur Ruby 2.7 und 3.2)
- Rust
- Scala(Scala3.2.2 oder früher)
- Shell
- TSX
- TypeScript (alle Versionen)

Unterstützte Sprachen für die Analyse der Softwarekomposition

- Go
- Java(Java25 oder früher)
- JavaScript(EMCMAScript 2021 oder früher)
- PHP(PHP8.2 oder früher)
- Python(Python3.13 oder früher innerhalb der Serie Python 3)
- .Net

- Ruby(nur Ruby 2.7 und 3.2)
- Rust

> **Note**
>
> Für npm (Node.js) -Projekte identifiziert die Zusammensetzungsanalyse der Code-Sicherheitssoftware von Amazon Inspector Abhängigkeiten aus dem `package.json` Manifest und schließt die `package-lock.json` Sperrdatei von der Analyse aus. Abhängigkeiten, die nur in vorkommen`package-lock.json`, einschließlich transitiver Abhängigkeiten und der genauen Versionen, die damit verknüpft werden, werden nicht ausgewertet.

Sprachen für Infrastruktur als Code

- AWS CDK (PythonundTypeScript)
- CloudFormation (2010—09—09)
- Terraform(1.6.2 oder früher)

Deaktivierung der Codesicherheit

Weitere Informationen zur Deaktivierung von Code Security finden Sie unter Deaktivieren eines Scantyps.

Die Ergebnisse von Amazon Inspector verstehen

Amazon Inspector generiert ein Ergebnis, wenn es eine Sicherheitslücke mit einem Fix oder einem ausstehenden Fix in Amazon EC2 EC2-Instances, Amazon ECR-Containern, Images und Lambda-Funktionen entdeckt. Es generiert auch Ergebnisse für Code-Schwachstellen, die im Quellcode von Erstanbieteranwendungen, in Abhängigkeiten von Drittanbieteranwendungen und in Infrastructure as Code entdeckt wurden. Ein Ergebnis ist ein detaillierter Bericht über eine Sicherheitslücke, die sich auf eine Ihrer AWS Ressourcen auswirkt.

Die Ergebnisse sind nach Sicherheitslücken benannt und enthalten Bewertungen des Schweregrads, Informationen zu betroffenen AWS Ressourcen und Ressourcen, die nicht zur AWS Verfügung stehen, sowie Einzelheiten zur Behebung erkannter Sicherheitslücken. Amazon Inspector speichert all Ihre aktiven Ergebnisse, bis Sie sie korrigieren.

Wenn eine Ressource gelöscht, beendet oder nicht mehr gescannt werden kann, schließt Amazon Inspector automatisch die mit der Ressource verknüpften Ergebnisse und löscht die Ergebnisse dann nach 3 Tagen. Wenn Ergebnisse aus einem anderen Grund geschlossen werden, werden sie nach 30 Tagen gelöscht.

> **Note**
>
> Amazon Inspector öffnet ein behobenes Ergebnis innerhalb von sieben Tagen nach Abschluss des Fundes erneut, falls das Problem, das die Sicherheitsanfälligkeit verursacht hat, erneut auftritt.

Wenn Sie Amazon Inspector deaktivieren, werden die Ergebnisse nach 24 Stunden entfernt. Wenn eine Ressource beendet wird, werden alle Ergebnisse, die sich auf die Ressource beziehen, nach 3 Tagen entfernt. Das Gleiche gilt für alle Ergebnisse, die mit einer Ressource verknüpft sind und für die das Scannen nicht mehr zulässig ist. Wenn AWS Ihr Konto gesperrt wird, werden die Ergebnisse nach 90 Tagen entfernt. Die Ergebnisse für gestoppte Instances bleiben aktiv.

Die Ergebnisse besagen

Amazon Inspector kategorisiert Ergebnisse in die folgenden Bundesstaaten.

Aktiv

Amazon Inspector stuft ein Ergebnis, das nicht behoben wurde, als Aktiv ein.

Unterdrückt

Amazon Inspector stuft ein Ergebnis, das einer oder mehreren Unterdrückungsregeln unterliegt, als Unterdrückt ein.

Closed (Abgeschlossen)

Wenn ein Ergebnis behoben wurde, stuft Amazon Inspector das Ergebnis als geschlossen ein.

Themen

- Suchtypen von Amazon Inspector
- Ergebnisse von Amazon Inspector anzeigen
- Details zu Ihren Amazon Inspector Inspector-Ergebnissen anzeigen
- Den Amazon Inspector-Score anzeigen und Details zur Schwachstellenanalyse verstehen
- Grundlegendes zu den Schweregraden Ihrer Amazon Inspector Inspector-Ergebnisse

Suchtypen von Amazon Inspector

In diesem Abschnitt werden die verschiedenen Findetypen in Amazon Inspector beschrieben.

Themen

- Sicherheitslücke im Package
- Sicherheitslücke im Code
- Erreichbarkeit über das Netzwerk

Sicherheitslücke im Package

Package von Ergebnissen zu Sicherheitslücken in Paketen werden Softwarepakete in Ihrer AWS Umgebung identifiziert, die häufig auftretenden Sicherheitslücken (CVEs) ausgesetzt sind. Angreifer können diese ungepatchten Sicherheitslücken ausnutzen, um die Vertraulichkeit, Integrität oder Verfügbarkeit von Daten zu gefährden oder auf andere Systeme zuzugreifen. Das CVE-System ist eine Referenzmethode für öffentlich bekannte Sicherheitslücken und Sicherheitslücken. Weitere Informationen finden Sie unter https://www.cve.org/.

Amazon Inspector kann Ergebnisse zu Sicherheitslücken in Paketen für EC2-Instances, ECR-Container-Images und Lambda-Funktionen generieren. Die Ergebnisse der Sicherheitslücken in

Paketen enthalten Details, die für diese Art von Entdeckung einzigartig sind. Bei diesen Details handelt es sich um den Inspector-Score und die Schwachstelleninformationen.

Bei Windows EC2-Instances können gefundene Sicherheitslücken in Paketen von der Microsoft Knowledge Base (KB) IDs statt von einzelnen CVEs Instanzen identifiziert werden. Wenn ein KB-Update eine oder mehrere behandelt CVEs, meldet Amazon Inspector beispielsweise `KB5023697` einen einzelnen KB-Befund anstelle eines separaten Ergebnisses für jeden CVE. Ein KB-Ergebnis gibt den höchsten CVSS-Score, EPSS-Score und die höchste Exploit-Verfügbarkeit für alle Komponenten an. CVEs

Sicherheitslücke im Code

Die Ergebnisse von Sicherheitslücken im Code helfen dabei, Codezeilen zu identifizieren, die ausgenutzt werden können. Zu den Sicherheitslücken im Code gehören fehlende Verschlüsselung, Datenlecks, Injektionsfehler und schwache Kryptografie. Amazon Inspector generiert mithilfe von Lambda-Funktionsscans und seiner Code Security-Funktion Erkenntnisse zu Sicherheitslücken im Code.

Amazon Inspector bewertet den Anwendungscode der Lambda-Funktion mithilfe automatisierter Argumentation und maschinellem Lernen, um den Anwendungscode auf allgemeine Sicherheitsbestimmungen hin zu analysieren. Es identifiziert Richtlinienverstöße und Sicherheitslücken auf der Grundlage interner Detektoren, die in Zusammenarbeit mit Amazon Q entwickelt wurden. Eine Liste möglicher Erkennungen finden Sie in der Amazon Q Detector Library.

Beim Codescannen werden Codefragmente erfasst, um erkannte Sicherheitslücken hervorzuheben. Beispielsweise kann ein Codeausschnitt hartcodierte Anmeldeinformationen oder andere vertrauliche Materialien im Klartext enthalten. Amazon Q speichert Codefragmente, die mit Code-Schwachstellen verknüpft sind. Standardmäßig ist Ihr Code mit einem AWS eigenen Schlüssel verschlüsselt. Sie können jedoch einen vom Kunden verwalteten Schlüssel erstellen, um Ihren Code zu verschlüsseln, wenn Sie mehr Kontrolle über diese Informationen haben möchten. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand für den Code in Ihren Ergebnissen.

> **Note**
>
> Der delegierte Administrator einer Organisation kann keine Codefragmente einsehen, die zu Mitgliedskonten gehören.

Erreichbarkeit über das Netzwerk

Die Ergebnisse der Netzwerkerreichbarkeit deuten darauf hin, dass es in Ihrer Umgebung offene Netzwerkpfade zu Amazon EC2 EC2-Instances gibt. Diese Ergebnisse treten auf, wenn Ihre TCP- und UDP-Ports von den VPC-Edges aus erreichbar sind, z. B. ein Internet-Gateway (einschließlich Instances hinter Application Load Balancers oder Classic Load Balancers), eine VPC-Peering-Verbindung oder ein VPN über ein virtuelles Gateway. Diese Ergebnisse heben Netzwerkkonfigurationen hervor, die möglicherweise zu freizügig sind, wie z. B. schlecht verwaltete Sicherheitsgruppen, Zugriffskontrolllisten oder Internet-Gateways, oder die potenziell böswilligen Zugriff ermöglichen.

Amazon Inspector generiert nur Ergebnisse zur Netzwerkerreichbarkeit für Amazon EC2 EC2-Instances. Amazon Inspector führt alle 12 Stunden Scans nach Ergebnissen der Netzwerkerreichbarkeit durch, sobald Amazon Inspector aktiviert ist.

Amazon Inspector bewertet beim Scannen nach Netzwerkpfaden die folgenden Konfigurationen:

- Amazon EC2-Instances
- Application Load Balancer
- Direct Connect
- Elastic Load Balancers
- Elastic-Network-Schnittstellen
- Internet-Gateways
- Listen zur Netzwerkzugriffskontrolle
- Routing-Tabellen
- Sicherheitsgruppen
- Subnets
- Virtuelle private Clouds
- Virtuelle private Gateways
- VPC-Endpunkte
- VPC-Gateway-Endpunkte
- VPC-Peering-Verbindungen
- VPN-Verbindungen

Ergebnisse von Amazon Inspector anzeigen

Sie können die Ergebnisse in der Amazon Inspector Inspector-Konsole und mit der Amazon Inspector `ListFindings`Inspector-API anzeigen. In der Amazon Inspector Inspector-Konsole können Sie alle Ihre Ergebnisse auf den Bildschirmen Dashboard und Ergebnisse einsehen. Standardmäßig werden auf diesen Bildschirmen nur Ihre aktiven und kritischen Ergebnisse angezeigt. Sie können jedoch nach Ergebnissen filtern oder sich die Ergebnisse nach Kategorien anzeigen lassen. Sie können einige Ergebnisse auch in Security Hub CSPM und Amazon ECR einsehen, wenn Sie diese Integrationen aktivieren. Die Verfahren in diesem Abschnitt beschreiben, wie Ergebnisse in der Amazon Inspector Inspector-Konsole und mit der Amazon Inspector `ListFindings` Inspector-API angezeigt werden.

Console

Um die Ergebnisse von Amazon Inspector einzusehen

1. Melden Sie sich mit Ihren Zugangsdaten an. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. (Optional) Wählen Sie im Navigationsbereich Dashboard aus. Das Dashboard zeigt einen Überblick über den Versicherungsschutz für Ihre Umgebung und nur Ihre aktiven und kritischen Ergebnisse.
3. (Optional) Wählen Sie im Navigationsbereich Findings aus. Auf diesem Bildschirm werden alle Ihre aktiven Ergebnisse aufgeführt. Sie können Filterkriterien verwenden, um bestimmte Ergebnisse anzuzeigen. Um Ergebnisse aus der Liste auszuschließen, erstellen Sie eine Unterdrückungsregel. Um Details zu einem Ergebnis anzuzeigen, wählen Sie den Namen des Ergebnisses.
4. (Optional) Wählen Sie im Navigationsbereich eine der folgenden Optionen, um Ihre Ergebnisse nach Kategorien geordnet anzuzeigen:

   - Nach Sicherheitslücke — Zeigt Sicherheitslücken mit den kritischsten Ergebnissen an.
   - Nach Konto — Zeigt Konten mit den kritischsten Ergebnissen an. Diese Kategorie steht nur delegierten Administratoren zur Verfügung.
   - Nach Instanz — Zeigt Amazon EC2 EC2-Instances mit den kritischsten Ergebnissen an. Diese Kategorie enthält keine Informationen zur Netzwerkverfügbarkeit.
   - Nach Container-Image — Zeigt Amazon ECR-Container-Images mit den wichtigsten Ergebnissen an. Diese Kategorie enthält auch grundlegende Informationen zu Ihren Container-Images. Es enthält sogar Details, z. B. wie viele Amazon ECS-Aufgaben und

Amazon EKS-Pods bereitgestellt werden. Auf diesem Bildschirm können Sie herausfinden, wie viele in den letzten 24 Stunden ausgeführt und gestoppt tasks/pods wurden.

- Nach Container-Repository — Zeigt Container-Repositorys mit den wichtigsten Ergebnissen an.
- Nach Lambda-Funktion — Zeigt Lambda-Funktionen mit den wichtigsten Ergebnissen an.

API

Um die Ergebnisse von Amazon Inspector einzusehen

- Führen Sie den ListFindingsAPI-Vorgang aus. Geben Sie in der Anfrage FilterCriteria an, um bestimmte Ergebnisse zurückzugeben.

Details zu Ihren Amazon Inspector Inspector-Ergebnissen anzeigen

Das Verfahren in diesem Abschnitt beschreibt, wie Sie Details zu den Ergebnissen von Amazon Inspector anzeigen können.

Um die Details zu einem Ergebnis anzuzeigen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home
2. Wählen Sie die Region aus, in der Sie sich die Ergebnisse ansehen möchten.
3. Wählen Sie im Navigationsbereich Findings aus, um die Ergebnisliste anzuzeigen
4. (Optional) Verwenden Sie die Filterleiste, um ein bestimmtes Ergebnis auszuwählen. Weitere Informationen finden Sie unter Filtern Ihrer Amazon Inspector Inspector-Ergebnisse.
5. Wählen Sie ein Ergebnis aus, um das zugehörige Detailfenster anzuzeigen.

Der Bereich mit den Ergebnisdetails enthält die grundlegenden Erkennungsmerkmale des Ergebnisses. Dazu gehören der Titel des Befundes sowie eine grundlegende Beschreibung der identifizierten Sicherheitslücke, Vorschläge zur Behebung und eine Bewertung des Schweregrads. Informationen zur Bewertung finden Sie unterGrundlegendes zu den Schweregraden Ihrer Amazon Inspector Inspector-Ergebnisse.

Die für ein Ergebnis verfügbaren Details variieren je nach Art des Ergebnisses und der betroffenen Ressource.

Alle Ergebnisse enthalten die AWS-Konto ID-Nummer, für die das Ergebnis identifiziert wurde, einen Schweregrad, einen Befundtyp, das Datum, an dem das Ergebnis erstellt wurde, und einen Abschnitt „Betroffene Ressource“ mit Details zu dieser Ressource.

Der Typ des Ergebnisses bestimmt, welche Informationen zur Problembehebung und zur Schwachstellenanalyse für das Ergebnis verfügbar sind. Je nach Art des Ergebnisses sind unterschiedliche Ergebnisdetails verfügbar.

Sicherheitslücke im Package

Ergebnisse zu Sicherheitslücken in Paketen sind für EC2-Instances, ECR-Container-Images und Lambda-Funktionen verfügbar. Weitere Informationen finden Sie unter Sicherheitslücke im Package.

Zu den Ergebnissen der Paketschwachstelle gehören auchDen Amazon Inspector-Score anzeigen und Details zur Schwachstellenanalyse verstehen.

Dieser Befundtyp enthält die folgenden Details:

- Fix verfügbar — Zeigt an, ob die Sicherheitsanfälligkeit in einer neueren Version der betroffenen Pakete behoben wurde. Hat einen der folgenden Werte:
  - `YES`, was bedeutet, dass alle betroffenen Pakete eine feste Version haben.
  - `NO`, was bedeutet, dass keine betroffenen Pakete eine feste Version haben.
  - `PARTIAL`, was bedeutet, dass eines oder mehrere (aber nicht alle) der betroffenen Pakete eine feste Version haben.
- Exploit verfügbar — Zeigt an, dass es sich bei der Sicherheitsanfälligkeit um einen bekannten Exploit handelt.
  - `YES`, was bedeutet, dass es sich bei der in Ihrer Umgebung entdeckten Sicherheitslücke um einen bekannten Exploit handelt. Amazon Inspector hat keinen Einblick in die Verwendung von Exploits in einer Umgebung.
  - `NO`, was bedeutet, dass für diese Sicherheitsanfälligkeit kein Exploit bekannt ist.
- Betroffene Pakete — Listet jedes Paket auf, das bei der Entdeckung als gefährdet identifiziert wurde, sowie die Einzelheiten zu jedem Paket:
- Dateipfad — Die EBS-Volume-ID und die Partitionsnummer, die mit einem Ergebnis verknüpft sind. Dieses Feld ist in den Ergebnissen für EC2-Instances enthalten, die mit gescannt wurden. Scannen ohne Agenten
- Installierte Version//Behobene Version — Die Versionsnummer des aktuell installierten Pakets, für das eine Sicherheitslücke erkannt wurde. Vergleichen Sie die installierte Versionsnummer

mit dem Wert nach dem Schrägstrich (/). Der zweite Wert ist die Versionsnummer des Pakets, das die entdeckte Sicherheitsanfälligkeit behebt, wie sie in den allgemeinen Sicherheitslücken (CVEs) oder in der zugehörigen Empfehlung angegeben ist. Wenn die Sicherheitsanfälligkeit in mehreren Versionen behoben wurde, wird in diesem Feld die neueste Version aufgeführt, die den Fix enthält. Wenn ein Fix nicht verfügbar ist, ist dieser Wert`None available`.

> **Note**
>
> Wenn ein Ergebnis erkannt wurde, bevor Amazon Inspector begann, dieses Feld in die Ergebnisse aufzunehmen, ist der Wert für dieses Feld leer. Möglicherweise ist jedoch ein Fix verfügbar.

- Paketmanager — Der Paketmanager, der zur Konfiguration dieses Pakets verwendet wurde.
- Behebung — Die Behebungsbefehle variieren je nach Systemkonfiguration. Genauere Hinweise finden Sie in den Referenzen oder in der Dokumentation zum Betriebssystem. Sofern verfügbar, sollten Kunden die betroffenen Pakete auf die bereitgestellte fehlerfreie Version (oder eine neuere) aktualisieren.
- Details zur Sicherheitslücke — bietet einen Link zur bevorzugten Quelle von Amazon Inspector für das im Ergebnis identifizierte CVE, z. B. National Vulnerability Database (NVD), REDHAT oder ein anderer Betriebssystemanbieter. Darüber hinaus finden Sie die Schweregrade für den Befund. Weitere Informationen zur Bewertung des Schweregrads finden Sie beispielsweise unterGrundlegendes zu den Schweregraden Ihrer Amazon Inspector Inspector-Ergebnisse. Die folgenden Werte sind enthalten, einschließlich der jeweiligen Bewertungsvektoren:
  - Ergebnis des Exploit Prediction Scoring Systems (EPSS)
  - Ergebnis des Inspector
  - CVSS 3.1 von Amazon CVE
  - CVSS 3.1 von NVD
  - CVSS 2.0 von NVD (falls zutreffend, für ältere Versionen) CVEs
- Verwandte Sicherheitslücken — Spezifiziert weitere Sicherheitslücken im Zusammenhang mit der Entdeckung. In der Regel handelt es sich CVEs dabei um andere, die sich auf dieselbe Paketversion auswirken, oder um andere CVEs , die zur selben Gruppe gehören wie das gefundene CVE, je nach Angabe des Herstellers.
- Betroffene Ressourcen — Beinhaltet Informationen über die Registrierung, das Repository, den Ressourcentyp, die Image-ID und das Image-Betriebssystem. Es enthält auch Informationen, z. B. wann ein Image zuletzt übertragen wurde, wie viele Amazon ECS-Aufgaben und Amazon

EKS-Pods bereitgestellt wurden und wann das Image in den letzten 24 Stunden zuletzt verwendet wurde. Wenn Sie Amazon ECS-Aufgaben und Amazon EKS-Pods bereitgestellt haben, können Sie Details anzeigen, indem Sie den Wert für das Feld auswählen. Dadurch gelangen Sie zu einem Bildschirm, auf dem Sie Informationen wie den Cluster-ARN, wann die Ressource in den letzten 24 Stunden zuletzt verwendet wurde, die Anzahl der ausgeführten und gestoppten Ressourcen sowie den Namen und Typ der Arbeitslast anzeigen können.

Sicherheitslücke im Code

Die Ergebnisse von Sicherheitslücken im Code sind nur für Lambda-Funktionen verfügbar. Weitere Informationen finden Sie unter Sicherheitslücke im Code. Dieser Erkennungstyp enthält die folgenden Details:

- Fix verfügbar — Bei Code-Schwachstellen gilt dieser Wert immer`YES`.
- Name des Melders — Der Name des Amazon Q-Detektors, der zur Erkennung der Code-Sicherheitslücke verwendet wurde. Eine Liste möglicher Erkennungen finden Sie in der Q Detector Library.
- Melder-Tags — Die mit dem Detektor verknüpften Amazon Q-Tags. Amazon Q verwendet Tags, um Erkennungen zu kategorisieren.
- Relevante CWE — IDs der Common Weakness Enumeration (CWE), die mit der Sicherheitslücke im Code verknüpft sind.
- Dateipfad — Der Speicherort der Code-Sicherheitslücke.
- Ort der Sicherheitslücke — Für Sicherheitslücken beim Scannen von Lambda-Code zeigt dieses Feld die genauen Codezeilen an, in denen Amazon Inspector die Sicherheitsanfälligkeit gefunden hat.
- Vorgeschlagene Behebung — Hier wird vorgeschlagen, wie der Code bearbeitet werden kann, um das Problem zu beheben.

Erreichbarkeit über das Netzwerk

Ergebnisse zur Netzwerkerreichbarkeit sind nur für EC2-Instances verfügbar. Weitere Informationen finden Sie unter Erreichbarkeit über das Netzwerk. Dieser Ergebnistyp enthält die folgenden Details:

- Offener Portbereich — Der Portbereich, über den auf die EC2-Instance zugegriffen werden konnte.
- Offene Netzwerkpfade — Zeigt den offenen Zugriffspfad zur EC2-Instance an. Wählen Sie ein Element im Pfad aus, um weitere Informationen zu erhalten.
- Behebung — Empfiehlt eine Methode zum Schließen des offenen Netzwerkpfads.

Den Amazon Inspector-Score anzeigen und Details zur Schwachstellenanalyse verstehen

Amazon Inspector erstellt eine Bewertung für die Ergebnisse der Amazon Elastic Compute Cloud (Amazon EC2) -Instance. Sie können den Amazon Inspector-Score und die Informationen zu Sicherheitslücken in der Amazon Inspector Inspector-Konsole einsehen. Der Amazon Inspector Score bietet Ihnen Details, die Sie mit den Kennzahlen im Common Vulnerability Scoring System vergleichen können. Diese Details sind nur für die Entdeckung von Sicherheitslücken in Paketen verfügbar. In diesem Abschnitt wird beschrieben, wie Sie den Amazon Inspector-Score interpretieren und Informationen zu Sicherheitslücken verstehen.

Amazon Inspector-Punktzahl

Amazon Inspector erstellt für jedes Amazon EC2 EC2-Ergebnis eine Bewertung. Amazon Inspector ermittelt das Ergebnis, indem es die Informationen zur CVSS-Basisbewertung mit Informationen aus Ihrer Computerumgebung korreliert, z. B. Daten zur Netzwerkerreichbarkeit und Ausnutzbarkeit. Amazon Inspector unterstützt Amazon-, Debian- und RHEL-Anbieter. Jeder Anbieter stellt eine CVSS v3.1-Basisbewertung zur Verfügung. Für andere Anbieter verwendet Amazon Inspector einen CVSS-Basiswert, der von der National Vulnerability Database (NVD) bereitgestellt wird.

Aufgrund der FedRAMP-Anforderungen verwendet Amazon Inspector den CVSS v3.1-Basiswert als Standardscore. Ein CVSS 4.0-Basiswert wird jedoch in Ihre Schwachstellen-Metadaten aufgenommen, sofern ein solcher verfügbar ist. Die CVSS 4.0-Basisbewertung bietet zusätzliche Kennzahlen zur Verbesserung der Schwachstellenbewertung. Quelle und Version einer CVSS-Basisbewertung finden Sie in den Schwachstellendetails für einen Befund und in den exportierten Ergebnissen.

> **Note**
>
> Der Amazon Inspector Score ist für Linux-Instances, auf denen Ubuntu ausgeführt wird, nicht verfügbar. Ubuntu verwendet ein benutzerdefiniertes System zur Bewertung des Schweregrads, das sich von den CVSS-Werten unterscheidet.

Einzelheiten zur Amazon Inspector-Punktzahl

Wenn Sie die Detailseite eines Befundes öffnen, können Sie die Registerkarte Inspector Score und Vulnerability Intelligence auswählen. Dieses Feld zeigt den Unterschied zwischen dem Basiswert

und dem Inspector-Score. In diesem Abschnitt wird erklärt, wie Amazon Inspector den Schweregrad auf der Grundlage einer Kombination aus dem Amazon Inspector-Score und dem Hersteller-Score für das Softwarepaket zugewiesen hat. Wenn sich die Punktzahlen unterscheiden, wird in diesem Bereich erklärt, warum.

Im Abschnitt CVSS-Score-Metriken finden Sie eine Tabelle mit Vergleichen zwischen den CVSS-Basisscore-Metriken und dem Inspector-Score. Bei den verglichenen Kennzahlen handelt es sich um die Basiskennzahlen, die im CVSS-Spezifikationsdokument definiert sind, das von verwaltet wird. first.org Im Folgenden finden Sie eine Zusammenfassung der Basiskennzahlen:

Angriffsvektor

Der Kontext, in dem eine Sicherheitslücke ausgenutzt werden kann. Bei Ergebnissen von Amazon Inspector kann dies „Netzwerk“, „Angrenzendes Netzwerk“ oder „Lokal“ sein.

Komplexität des Angriffs

Dies beschreibt den Schwierigkeitsgrad, mit dem ein Angreifer konfrontiert sein wird, wenn er die Sicherheitsanfälligkeit ausnutzt. Eine niedrige Punktzahl bedeutet, dass der Angreifer nur wenige oder keine zusätzlichen Bedingungen erfüllen muss, um die Sicherheitsanfälligkeit auszunutzen. Eine hohe Punktzahl bedeutet, dass ein Angreifer erhebliche Anstrengungen unternehmen muss, um einen erfolgreichen Angriff mit dieser Sicherheitsanfälligkeit durchzuführen.

Privileg erforderlich

Dies beschreibt die Rechtestufe, die ein Angreifer benötigt, um eine Sicherheitslücke auszunutzen.

Interaktion mit dem Benutzer

Diese Metrik gibt an, ob für einen erfolgreichen Angriff unter Ausnutzung dieser Sicherheitsanfälligkeit ein menschlicher Benutzer erforderlich ist, der nicht der Angreifer ist.

Scope

Dies gibt an, ob sich eine Sicherheitsanfälligkeit in einer anfälligen Komponente auf Ressourcen in Komponenten auswirkt, die über den Sicherheitsbereich der anfälligen Komponente hinausgehen. Wenn dieser Wert Unverändert ist, sind die betroffene Ressource und die betroffene Ressource identisch. Wenn dieser Wert geändert ist, kann die anfällige Komponente ausgenutzt werden, um Ressourcen zu beeinträchtigen, die von verschiedenen Sicherheitsbehörden verwaltet werden.

Vertraulichkeit

Dabei wird das Ausmaß der Auswirkungen auf die Vertraulichkeit von Daten innerhalb einer Ressource gemessen, wenn die Sicherheitsanfälligkeit ausgenutzt wird. Dies reicht von „Keine“, bei der keine Vertraulichkeit verloren geht, bis hin zu „Hoch“, bei der alle Informationen innerhalb einer Ressource weitergegeben werden oder vertrauliche Informationen wie Passwörter oder Verschlüsselungsschlüssel preisgegeben werden können.

Integrität

Dabei wird das Ausmaß der Auswirkungen auf die Integrität der Daten innerhalb der betroffenen Ressource gemessen, wenn die Sicherheitsanfälligkeit ausgenutzt wird. Die Integrität ist gefährdet, wenn der Angreifer Dateien innerhalb der betroffenen Ressourcen verändert. Die Bewertung reicht von „Keine“, wobei der Angriff es einem Angreifer nicht ermöglicht, Informationen zu ändern, bis hin zu „Hoch“, bei dem die Sicherheitsanfälligkeit es einem Angreifer ermöglichen würde, einige oder alle Dateien zu ändern, oder die Dateien, die geändert werden könnten, schwerwiegende Folgen haben könnten.

Verfügbarkeit

Damit wird das Ausmaß der Auswirkungen auf die Verfügbarkeit der betroffenen Ressource gemessen, wenn die Sicherheitsanfälligkeit ausgenutzt wird. Die Bewertung reicht von „Keine“, wenn die Sicherheitsanfälligkeit die Verfügbarkeit überhaupt nicht beeinträchtigt, bis hin zu „Hoch“, bei dem der Angreifer bei Ausnutzung die Verfügbarkeit der Ressource vollständig verweigern oder dafür sorgen kann, dass ein Dienst nicht verfügbar ist.

Informationen zu Sicherheitslücken

In diesem Abschnitt werden die verfügbaren Informationen über das CVE von Amazon sowie branchenübliche Sicherheitsinformationsquellen wie die Cybersecurity and Infrastructure Security Agency (CISA) zusammengefasst.

> **Note**
>
> Intel von CISA oder Amazon wird nicht für alle CVEs verfügbar sein.

Sie können die Informationen zu Sicherheitslücken in der Konsole oder mithilfe der BatchGetFindingDetailsAPI einsehen. Die folgenden Details sind in der Konsole verfügbar:

ATT&CK

In diesem Abschnitt werden die Taktiken, Techniken und Verfahren (TTPs) von MITRE im Zusammenhang mit dem CVE beschrieben. Die zugehörigen TTPs werden angezeigt. Wenn mehr als zwei zutreffen, können TTPs Sie den Link auswählen, um eine vollständige Liste zu sehen. Wenn Sie eine Taktik oder Technik auswählen, werden Informationen zu dieser Taktik oder Technik auf der MITRE-Website geöffnet.

CISA

Dieser Abschnitt behandelt relevante Daten im Zusammenhang mit der Sicherheitsanfälligkeit. Das Datum, an dem die Cybersecurity and Infrastructure Security Agency (CISA) die Sicherheitslücke in den Katalog der bekannten Sicherheitslücken aufgenommen hat, basierend auf Hinweisen auf eine aktive Ausnutzung, und das Fälligkeitsdatum, an dem die CISA erwartet, dass die Systeme gepatcht werden. Diese Informationen stammen von CISA.

Bekannte Schadsoftware

In diesem Abschnitt sind bekannte Exploit-Kits und Tools aufgeführt, die diese Sicherheitsanfälligkeit ausnutzen.

Letztes Mal gemeldet

In diesem Abschnitt wird das Datum des letzten bekannten öffentlichen Exploits für diese Sicherheitsanfälligkeit angezeigt.

Grundlegendes zu den Schweregraden Ihrer Amazon Inspector Inspector-Ergebnisse

Wenn Amazon Inspector ein Ergebnis generiert, weist es dem Ergebnis einen Schweregrad zu. Schweregrade helfen Ihnen dabei, Ihre Ergebnisse zu bewerten und zu priorisieren. Der Schweregrad eines Ergebnisses entspricht einem numerischen Wert und einer Stufe: informativ, niedrig, mittel, hoch und kritisch. Amazon Inspector bestimmt den Schweregrad für ein Ergebnis auf der Grundlage des Ergebnistyps. In diesem Abschnitt wird beschrieben, wie Amazon Inspector für jeden Befundtyp eine Schweregradbewertung ermittelt.

Schweregrad der Sicherheitslücke im Softwar

Amazon Inspector verwendet die NVD/CVSS Bewertung als Grundlage für die Bewertung des Schweregrads von Sicherheitslücken in Softwarepaketen. Bei der NVD/CVSS Bewertung handelt

es sich um den Schweregrad der Sicherheitslücke, der vom NVD veröffentlicht und vom CVSS definiert wurde. Die NVD/CVSS Bewertung setzt sich aus Sicherheitsmetriken wie der Komplexität des Angriffs, dem Reifegrad des Exploit-Codes und den erforderlichen Rechten zusammen. Amazon Inspector erstellt eine numerische Bewertung von 1 bis 10, die den Schweregrad der Sicherheitsanfälligkeit widerspiegelt. Amazon Inspector stuft dies als Basiswert ein, da er den Schweregrad einer Sicherheitslücke anhand ihrer intrinsischen Merkmale widerspiegelt, die im Laufe der Zeit konstant sind. Bei dieser Bewertung wird auch davon ausgegangen, dass die Auswirkungen im schlimmsten Fall auf verschiedene bereitgestellte Umgebungen angemessen sind. Der CVSS v3-Standard ordnet die CVSS-Werte den folgenden Schweregraden zu.

Ergebnis	Bewertung
0	Informativ
0,1—3,9	Niedrig
4,0—6,9	Mittel
7,0—8,9	Hoch
9,0—10,0	Kritisch

Die gefundenen Sicherheitslücken in Paketen können auch den Schweregrad Untriaged haben. Das bedeutet, dass der Anbieter noch keinen Schwachstellen-Score für die entdeckte Sicherheitslücke festgelegt hat. In diesem Fall empfehlen wir, die Referenz URLs für das Ergebnis zu verwenden, um diese Sicherheitsanfälligkeit zu untersuchen und entsprechend zu reagieren.

Zu den Ergebnissen der Paketschwachstellen gehören die folgenden Bewertungen und die zugehörigen Bewertungsvektoren als Teil der Ergebnisdetails:

- EPSS-Punktzahl
- Ergebnis des Inspector
- CVSS 3.1 von Amazon CVE
- CVSS 3.1 von NVD
- CVSS 2.0 von NVD (falls zutreffend)

Schweregrad der Sicherheitslücke

Für die Entdeckung von Sicherheitslücken im Code verwendet Amazon Inspector die Schweregrade, die von den Amazon Q-Detektoren definiert wurden, die den Befund generiert haben. Jedem Detektor wird mithilfe des CVSS v3-Bewertungssystems ein Schweregrad zugewiesen.?

Schweregrad der Erreichbarkeit des Netzwerks

Amazon Inspector bestimmt den Schweregrad einer Sicherheitslücke im Netzwerk auf der Grundlage der offengelegten Services, Ports und Protokolle sowie der Art des offenen Pfads. In der folgenden Tabelle werden diese Schweregrade definiert. Der Wert in der Spalte Bewertung offener Pfade steht für offene Pfade von virtuellen Gateways, Peering-Verbindungen und VPCs AWS Direct Connect Netzwerken. Für alle anderen exponierten Dienste, Ports und Protokolle wurde der Schweregrad „Information“ eingestuft.

Service	TCP-Ports	UDP-Anschlüsse	Bewertung des Internetpfads	Pfadbewertung öffnen
DHCP	67, 68, 546, 547	67, 68, 546, 547	Mittel	Informativ
Elasticsearch	9300, 9200	N/A	Mittel	Informativ
FTP	21	21	Hoch	Mittel
Global Catalog LDAP	3268	N/A	Mittel	Informativ
Global Catalog LDAP über TLS	3269	N/A	Mittel	Informativ
HTTP	80	80	Niedrig	Informativ
HTTPS	443	443	Niedrig	Informativ
Kerberos	88, 464, 543, 544, 749, 751	88, 464, 749, 750, 751, 752	Mittel	Informativ
LDAP	389	389	Mittel	Informativ
LDAP über TLS	636	N/A	Mittel	Informativ

MongoDB	27017, 27018, 27019, 28017	N/A	Mittel	Informativ
MySQL	3306	N/A	Mittel	Informativ
NetBIOS	137, 139	137, 138	Mittel	Informativ
NFS	111, 2049, 4045, 1110	111, 2049, 4045, 1110	Mittel	Informativ
Oracle	1521, 1630	N/A	Mittel	Informativ
PostgreSQL	5432	N/A	Mittel	Informativ
Druckdienste	515	N/A	Hoch	Mittel
RDP	3389	3389	Mittel	Niedrig
RPC	111, 135, 530	111, 135, 530	Mittel	Informativ
SMB	445	445	Mittel	Informativ
SSH	22	22	Mittel	Niedrig
SQL Server	1433	1434	Mittel	Informativ
Syslog	601	514	Mittel	Informativ
Telnet	23	23	Hoch	Mittel
WINS	1512, 42	1512, 42	Mittel	Informativ

Ergebnisse in Amazon Inspector verwalten

Mit Amazon Inspector können Sie Ihre Ergebnisse auf unterschiedliche Weise verwalten. Sie können Ihre Ergebnisse nach ihrem Status filtern. Sie können Ihre Ergebnisse anhand von Filterkriterien durchsuchen. Sie können Unterdrückungsregeln erstellen, um Ergebnisse aus Ihrer Ergebnisliste auszuschließen. Sie können Ergebnisse auch nach AWS Security Hub CSPM Amazon und Amazon EventBridge Simple Storage Service (Amazon S3) exportieren.

Themen

- Filtern Ihrer Amazon Inspector Inspector-Ergebnisse
- Unterdrückung der Ergebnisse von Amazon Inspector
- Amazon Inspector-Ergebnisberichte exportieren
- Erstellung benutzerdefinierter Antworten auf Amazon Inspector-Ergebnisse mit Amazon EventBridge

Filtern Ihrer Amazon Inspector Inspector-Ergebnisse

Sie können Ihre Amazon Inspector Inspector-Ergebnisse mithilfe von Filterkriterien filtern. Wenn ein Ergebnis nicht Ihren Filterkriterien entspricht, schließt Amazon Inspector das Ergebnis aus der Ansicht aus. In diesem Abschnitt wird beschrieben, wie Sie Ihre Amazon Inspector Inspector-Ergebnisse mithilfe von Filterkriterien filtern können.

Filter in der Amazon Inspector Inspector-Konsole erstellen

In jeder Ergebnisansicht können Sie die Filterfunktion verwenden, um Ergebnisse mit bestimmten Merkmalen zu finden. Filter werden entfernt, wenn Sie zu einer anderen Ansicht mit Registerkarten wechseln.

Ein Filter besteht aus einem Filterkriterium, das aus einem Filterattribut und einem Filterwert besteht. Ergebnisse, die Ihren Filterkriterien nicht entsprechen, werden von der Ergebnisliste ausgeschlossen. Um beispielsweise alle Ergebnisse zu sehen, die mit Ihrem Administratorkonto verknüpft sind, können Sie das AWS Konto-ID-Attribut auswählen und es mit dem Wert Ihrer zwölfstelligen AWS Konto-ID verknüpfen.

Einige Filterkriterien gelten für alle Ergebnisse, während andere nur für bestimmte Ressourcentypen oder nur für Suchtypen verfügbar sind.

Um einen Filter auf die Ergebnisansicht anzuwenden

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Findings aus. In der Standardansicht werden alle Ergebnisse mit dem Status Aktiv angezeigt.
3. Um Ergebnisse nach Kriterien zu filtern, wählen Sie die Filterleiste Hinzufügen aus, um eine Liste aller zutreffenden Filterkriterien für diese Ansicht anzuzeigen. Verschiedene Filterkriterien sind in verschiedenen Ansichten verfügbar.
4. Wählen Sie aus der Liste ein Kriterium aus, nach dem Sie filtern möchten.
5. Geben Sie im Kriterieneingabebereich die gewünschten Filterwerte ein, um dieses Kriterium zu definieren.
6. Wählen Sie Anwenden, um dieses Filterkriterium auf Ihre aktuellen Ergebnisse anzuwenden. Sie können weitere Filterkriterien hinzufügen, indem Sie erneut die Filtereingabeleiste auswählen.
7. (Optional) Um Ihre unterdrückten oder geschlossenen Ergebnisse anzuzeigen, wählen Sie in der Filterleiste Aktiv und dann Unterdrückt oder Geschlossen aus. Wählen Sie „Alle anzeigen“, um aktive, unterdrückte und geschlossene Ergebnisse in derselben Ansicht anzuzeigen.

Unterdrückung der Ergebnisse von Amazon Inspector

Sie können Unterdrückungsregeln erstellen, um Ergebnisse auszublenden, die den Kriterien entsprechen. Sie können beispielsweise eine Unterdrückungsregel erstellen, um Ergebnisse anhand ihres Schweregrads auszublenden. Wenn Amazon Inspector ein Ergebnis generiert, das Ihrer Unterdrückungsregel entspricht, unterdrückt Amazon Inspector das Ergebnis und blendet es aus. Amazon Inspector speichert unterdrückte Ergebnisse, bis sie behoben sind. Sobald ein unterdrücktes Ergebnis behoben wurde, schließt Amazon Inspector das Ergebnis. Sie können unterdrückte Ergebnisse in der Konsole einsehen.

Sie erstellen Unterdrückungsregeln, um Ihre wichtigsten Ergebnisse zu priorisieren. Unterdrückungsregeln haben keine Auswirkungen auf Ihre Ergebnisse, da sie nur Ergebnisse vor dem Zugriff verbergen. Sie können keine Unterdrückungsregel erstellen, die Ergebnisse schließt oder behebt. Sie können unerwünschte Ergebnisse auch AWS Security Hub CSPM mit einer EventBridge Amazon-Regel unterdrücken. Die Verfahren in diesem Abschnitt beschreiben, wie Sie eine Unterdrückungsregel erstellen, anzeigen, bearbeiten und löschen.

> **Note**
>
> Nur der delegierte Administrator einer Organisation kann Unterdrückungsregeln erstellen und verwalten.

Eine Unterdrückungsregel erstellen

Sie können Unterdrückungsregeln erstellen, um die Liste der Ergebnisse zu filtern, die standardmäßig angezeigt werden. Sie können eine Unterdrückungsregel programmgesteuert erstellen, indem Sie die CreateFilterAPI verwenden und `SUPPRESS` als Wert für angeben. `action`

> **Note**
>
> Nur eigenständige Konten und delegierte Amazon Inspector-Administratoren können Unterdrückungsregeln erstellen und verwalten. Mitgliedern einer Organisation wird im Navigationsbereich keine Option für Unterdrückungsregeln angezeigt.

Um eine Unterdrückungsregel zu erstellen (Konsole)

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich die Option Suppression Rules aus. Wählen Sie dann Create rule (Regel erstellen) aus.
3. Gehen Sie für jedes Kriterium wie folgt vor:
   - Wählen Sie die Filterleiste aus, um eine Liste mit Filterkriterien anzuzeigen, die Sie zu Ihrer Unterdrückungsregel hinzufügen können.
   - Wählen Sie die Filterkriterien für Ihre Unterdrückungsregel aus.
4. Wenn Sie mit dem Hinzufügen von Kriterien fertig sind, geben Sie einen Namen für die Regel und optional eine Beschreibung ein.
5. Wählen Sie Save rule (Regel speichern). Amazon Inspector wendet die neue Unterdrückungsregel an und verbirgt alle Ergebnisse, die den Kriterien entsprechen. Beachten Sie, dass es bis zu 24 Stunden dauern kann, bis die Regeln zur Unterdrückung wirksam werden.

Unterdrückte Ergebnisse anzeigen

Standardmäßig zeigt Amazon Inspector keine unterdrückten Ergebnisse in der Amazon Inspector Inspector-Konsole an. Sie können sich jedoch die Ergebnisse ansehen, die durch eine bestimmte Regel unterdrückt wurden.

Um unterdrückte Ergebnisse anzuzeigen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Unterdrückungsregeln aus.
3. Wählen Sie in der Liste der Unterdrückungsregeln den Titel der Regel aus.

Eine Unterdrückungsregel bearbeiten

Sie können jederzeit Änderungen an den Unterdrückungsregeln vornehmen.

Um Unterdrückungsregeln zu ändern

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich die Option Suppression Rules aus.
3. Wählen Sie den Namen der Unterdrückungsregel aus, die Sie ändern möchten, und klicken Sie dann auf Bearbeiten.
4. Nehmen Sie die gewünschten Änderungen vor und wählen Sie dann Speichern.

Löschen einer Unterdrückungsregel

Sie können Unterdrückungsregeln löschen. Wenn Sie eine Unterdrückungsregel löschen, beendet Amazon Inspector die Unterdrückung neuer und vorhandener Ergebnisse, die die Regelkriterien erfüllen und nicht durch andere Regeln unterdrückt werden.

Nachdem Sie eine Unterdrückungsregel gelöscht haben, haben neue und bestehende Ergebnisse, die die Kriterien der Regel erfüllen, den Status Aktiv. Das bedeutet, dass sie standardmäßig auf der Amazon Inspector Inspector-Konsole angezeigt werden. Darüber hinaus veröffentlicht Amazon Inspector diese Ergebnisse im Rahmen von AWS Veranstaltungen an Security Hub CSPM und Amazon EventBridge .

Um eine Unterdrückungsregel zu löschen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Unterdrückungsregeln aus.
3. Aktivieren Sie das Kontrollkästchen neben dem Titel der Unterdrückungsregel, die Sie löschen möchten.
4. Wählen Sie Löschen und bestätigen Sie dann Ihre Auswahl, um die Regel dauerhaft zu löschen.

Amazon Inspector-Ergebnisberichte exportieren

Ein Ergebnisbericht ist eine CSV- oder JSON-Datei, die eine detaillierte Momentaufnahme Ihrer Ergebnisse bietet. Sie können einen Ergebnisbericht nach AWS Security Hub CSPM Amazon und Amazon EventBridge Simple Storage Service (Amazon S3) exportieren. Wenn Sie einen Ergebnisbericht konfigurieren, geben Sie an, welche Ergebnisse darin enthalten sein sollen. Standardmäßig enthält Ihr Ergebnisbericht Daten für alle Ihre aktiven Ergebnisse. Wenn Sie der delegierte Administrator für eine Organisation sind, enthält Ihr Ergebnisbericht Daten für alle Mitgliedskonten in der Organisation. Um einen Ergebnisbericht anzupassen, erstellen Sie einen Filter und wenden Sie ihn an.

Wenn Sie einen Ergebnisbericht exportieren, verschlüsselt Amazon Inspector Ihre Ergebnisdaten mit einer von AWS KMS key Ihnen angegebenen. Nachdem Amazon Inspector Ihre Ergebnisdaten verschlüsselt hat, speichert es Ihren Ergebnisbericht in einem von Ihnen angegebenen Amazon S3-Bucket. Ihr AWS KMS Schlüssel muss in derselben Weise AWS-Region wie Ihr Amazon S3-Bucket verwendet werden. Ihre AWS KMS Schlüsselrichtlinie muss Amazon Inspector die Verwendung ermöglichen, und Ihre Amazon S3-Bucket-Richtlinie muss Amazon Inspector das Hinzufügen von Objekten ermöglichen. Nachdem Sie Ihren Ergebnisbericht exportiert haben, können Sie ihn aus Ihrem Amazon S3-Bucket herunterladen oder an einen neuen Speicherort übertragen. Sie können Ihren Amazon S3-Bucket auch als Repository für andere exportierte Ergebnisberichte verwenden.

In diesem Abschnitt wird beschrieben, wie Sie einen Ergebnisbericht in die Amazon Inspector-Konsole exportieren. Für die folgenden Aufgaben müssen Sie Ihre Berechtigungen überprüfen, einen Amazon S3-Bucket konfigurieren und einen AWS KMS key Ergebnisbericht konfigurieren und exportieren.

> **Note**
>
> Wenn Sie einen Ergebnisbericht mit der Amazon CreateFindingsReport Inspector-API exportieren, können Sie nur Ihre aktiven Ergebnisse einsehen. Wenn Sie Ihre unterdrückten oder geschlossenen Ergebnisse anzeigen möchten, müssen Sie `SUPPRESSED` oder `CLOSED` als Teil Ihrer Filterkriterien angeben.

Aufgaben

- Schritt 1: Überprüfen Sie Ihre Berechtigungen
- Schritt 2: Konfigurieren Sie einen S3-Bucket
- Schritt 3: Konfigurieren Sie eine AWS KMS key
- Schritt 4: Konfigurieren und exportieren Sie einen Ergebnisbericht
- Beheben Sie Exportfehler

Schritt 1: Überprüfen Sie Ihre Berechtigungen

> **Note**
>
> Nachdem Sie zum ersten Mal einen Ergebnisbericht exportiert haben, sind die Schritte 1—3 optional. Das Befolgen dieser Schritte hängt davon ab, ob Sie denselben Amazon S3-Bucket verwenden möchten, und AWS KMS key für andere exportierte Ergebnisberichte. Wenn Sie einen Ergebnisbericht programmgesteuert exportieren möchten, nachdem Sie die Schritte 1—3 abgeschlossen haben, verwenden Sie die Amazon CreateFindingsReport Inspector-API.

Bevor Sie einen Ergebnisbericht aus Amazon Inspector exportieren, stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen verfügen, um Ergebnisberichte zu exportieren und Ressourcen zum Verschlüsseln und Speichern der Berichte zu konfigurieren. Um Ihre Berechtigungen zu überprüfen, verwenden Sie AWS Identity and Access Management (IAM), um die IAM-Richtlinien zu überprüfen, die Ihrer IAM-Identität zugeordnet sind. Vergleichen Sie dann die Informationen in diesen Richtlinien mit der folgenden Liste von Aktionen, die Sie ausführen müssen, um einen Ergebnisbericht zu exportieren.

Amazon Inspector

Stellen Sie für Amazon Inspector sicher, dass Sie die folgenden Aktionen ausführen dürfen:

- `inspector2:ListFindings`
- `inspector2:CreateFindingsReport`

Mit diesen Aktionen können Sie Ergebnisdaten für Ihr Konto abrufen und diese Daten in Ergebnisberichte exportieren.

Wenn Sie vorhaben, umfangreiche Berichte programmgesteuert zu exportieren, können Sie auch überprüfen, ob Sie die folgenden Aktionen ausführen dürfen:`inspector2:GetFindingsReportStatus`, um den Status von Berichten zu überprüfen und`inspector2:CancelFindingsReport`, um laufende Exporte abzubrechen.

AWS KMS

Stellen Sie beispielsweise sicher AWS KMS, dass Sie die folgenden Aktionen ausführen dürfen:

- `kms:GetKeyPolicy`
- `kms:PutKeyPolicy`

Mit diesen Aktionen können Sie die Schlüsselrichtlinie abrufen und aktualisieren AWS KMS key , die Amazon Inspector zur Verschlüsselung Ihres Berichts verwenden soll.

Um die Amazon Inspector-Konsole zum Exportieren eines Berichts zu verwenden, stellen Sie außerdem sicher, dass Sie die folgenden AWS KMS Aktionen ausführen dürfen:

- `kms:DescribeKey`
- `kms:ListAliases`

Mit diesen Aktionen können Sie Informationen über Ihr Konto abrufen und anzeigen. AWS KMS keys Sie können dann einen dieser Schlüssel auswählen, um Ihren Bericht zu verschlüsseln.

Wenn Sie vorhaben, einen neuen KMS-Schlüssel für die Verschlüsselung Ihres Berichts zu erstellen, müssen Sie auch berechtigt sein, die `kms:CreateKey` Aktion auszuführen.

Amazon S3

Stellen Sie für Amazon S3 sicher, dass Sie die folgenden Aktionen ausführen dürfen:

- `s3:CreateBucket`
- `s3:DeleteObject`

- `s3:PutBucketAcl`
- `s3:PutBucketPolicy`
- `s3:PutBucketPublicAccessBlock`
- `s3:PutObject`
- `s3:PutObjectAcl`

Mit diesen Aktionen können Sie den S3-Bucket erstellen und konfigurieren, in dem Amazon Inspector Ihren Bericht speichern soll. Sie ermöglichen es Ihnen auch, Objekte zum Bucket hinzuzufügen und daraus zu löschen.

Wenn Sie vorhaben, die Amazon Inspector-Konsole zum Exportieren Ihres Berichts zu verwenden, stellen Sie außerdem sicher, dass Sie die `s3:GetBucketLocation` Aktionen `s3:ListAllMyBuckets` und ausführen dürfen. Mit diesen Aktionen können Sie Informationen über die S3-Buckets für Ihr Konto abrufen und anzeigen. Sie können dann einen dieser Buckets auswählen, um den Bericht zu speichern.

Wenn Sie eine oder mehrere der erforderlichen Aktionen nicht ausführen dürfen, bitten Sie Ihren AWS Administrator um Unterstützung, bevor Sie mit dem nächsten Schritt fortfahren.

Schritt 2: Konfigurieren Sie einen S3-Bucket

Nachdem Sie Ihre Berechtigungen überprüft haben, können Sie den S3-Bucket konfigurieren, in dem Sie Ihren Ergebnisbericht speichern möchten. Es kann sich um einen vorhandenen Bucket für Ihr eigenes Konto oder um einen vorhandenen Bucket handeln, der einem anderen gehört AWS-Konto und auf den Sie zugreifen dürfen. Wenn Sie Ihren Bericht in einem neuen Bucket speichern möchten, erstellen Sie den Bucket, bevor Sie fortfahren.

Der S3-Bucket muss sich in AWS-Region den Ergebnisdaten befinden, die Sie exportieren möchten. Wenn Sie beispielsweise Amazon Inspector in der Region USA Ost (Nord-Virginia) verwenden und Ergebnisdaten für diese Region exportieren möchten, muss sich der Bucket auch in der Region USA Ost (Nord-Virginia) befinden.

Darüber hinaus muss die Richtlinie des Buckets Amazon Inspector das Hinzufügen von Objekten zum Bucket ermöglichen. In diesem Thema wird erklärt, wie die Bucket-Richtlinie aktualisiert wird, und es enthält ein Beispiel für die Anweisung, die der Richtlinie hinzugefügt werden muss. Detaillierte Informationen zum Hinzufügen und Aktualisieren von Bucket-Richtlinien finden Sie unter Verwenden von Bucket-Richtlinien im Amazon Simple Storage Service-Benutzerhandbuch.

Wenn Sie Ihren Bericht in einem S3-Bucket speichern möchten, der einem anderen Konto gehört, arbeiten Sie mit dem Besitzer des Buckets zusammen, um die Richtlinie des Buckets zu aktualisieren. Besorgen Sie sich auch den URI für den Bucket. Sie müssen diese URI eingeben, wenn Sie Ihren Bericht exportieren.

Um die Bucket-Richtlinie zu aktualisieren

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon S3-Konsole unter https://console.aws.amazon.com/s3.
2. Wählen Sie im Navigationsbereich die Option Buckets aus.
3. Wählen Sie den S3-Bucket aus, in dem Sie den Ergebnisbericht speichern möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Bucket-Richtlinie die Option Bearbeiten aus.
6. Kopieren Sie die folgende Beispielanweisung in Ihre Zwischenablage:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allow-inspector",
            "Effect": "Allow",
            "Principal": {
                "Service": "inspector2.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:inspector2:us-
east-1:111122223333:report/*"
                }
```

```
            }
        }
    ]
}
```

7. Fügen Sie im Bucket-Richtlinieneditor auf der Amazon S3-Konsole die vorherige Anweisung in die Richtlinie ein, um sie der Richtlinie hinzuzufügen.

   Wenn Sie die Anweisung hinzufügen, stellen Sie sicher, dass die Syntax gültig ist. Bucket-Richtlinien verwenden das JSON-Format. Das bedeutet, dass Sie vor oder nach der Anweisung ein Komma hinzufügen müssen, je nachdem, an welcher Stelle Sie die Anweisung zur Richtlinie hinzufügen. Wenn Sie die Anweisung als letzte Anweisung hinzufügen, fügen Sie nach der schließenden Klammer der vorhergehenden Anweisung ein Komma ein. Wenn Sie die Anweisung als erste Anweisung oder zwischen zwei vorhandenen Anweisungen hinzufügen, fügen Sie nach der schließenden Klammer der Anweisung ein Komma ein.

8. Aktualisieren Sie die Anweisung mit den richtigen Werten für Ihre Umgebung. Dabei gilt:

   - *amzn-s3-demo-bucket*ist der Name des Buckets.
   - *111122223333*ist die Konto-ID für Ihre AWS-Konto.
   - *Region*ist der, AWS-Region in dem Sie Amazon Inspector verwenden und Amazon Inspector erlauben möchten, Berichte zum Bucket hinzuzufügen. Zum Beispiel `us-east-1` für die Region USA Ost (Nord-Virginia).

   > **Note**
   >
   > Wenn Sie Amazon Inspector in einer manuell aktivierten Umgebung verwenden AWS-Region, fügen Sie dem Wert für das `Service` Feld auch den entsprechenden Regionalcode hinzu. Dieses Feld gibt den Amazon Inspector-Serviceprinzipal an. Wenn Sie beispielsweise Amazon Inspector in der Region Naher Osten (Bahrain) verwenden, die den Regionalcode hat`me-south-1`, `inspector2.amazonaws.com` ersetzen Sie ihn `inspector2.me-south-1.amazonaws.com` in der Abrechnung durch.

   Beachten Sie, dass die Beispielanweisung Bedingungen definiert, die zwei globale IAM-Bedingungsschlüssel verwenden:

- aws: SourceAccount — Diese Bedingung ermöglicht es Amazon Inspector, Berichte nur für Ihr Konto zum Bucket hinzuzufügen. Dadurch wird verhindert, dass Amazon Inspector Berichte für andere Konten zum Bucket hinzufügt. Insbesondere gibt die Bedingung an, welches Konto den Bucket für die in der `aws:SourceArn` Bedingung angegebenen Ressourcen und Aktionen verwenden kann.

  Um Berichte für weitere Konten im Bucket zu speichern, fügen Sie dieser Bedingung die Konto-ID für jedes weitere Konto hinzu. Beispiel:

  ```
  "aws:SourceAccount": ["111122223333","444455556666","123456789012"]
  ```

- aws: SourceArn — Diese Bedingung schränkt den Zugriff auf den Bucket auf der Grundlage der Quelle der Objekte ein, die dem Bucket hinzugefügt werden. Sie verhindert, dass andere AWS-Services Objekte zum Bucket hinzufügen. Es verhindert auch, dass Amazon Inspector dem Bucket Objekte hinzufügt, während andere Aktionen für Ihr Konto ausgeführt werden. Insbesondere ermöglicht die Bedingung Amazon Inspector, dem Bucket nur dann Objekte hinzuzufügen, wenn es sich bei den Objekten um Ergebnisberichte handelt, und nur, wenn diese Berichte von dem Konto und in der Region erstellt wurden, die in der Bedingung angegeben sind.

  Damit Amazon Inspector die angegebenen Aktionen für weitere Konten ausführen kann, fügen Sie Amazon Resource Names (ARNs) für jedes weitere Konto zu dieser Bedingung hinzu. Beispiel:

  ```
  "aws:SourceArn": [
      "arn:aws:inspector2:Region:111122223333:report/*",
      "arn:aws:inspector2:Region:444455556666:report/*",
      "arn:aws:inspector2:Region:123456789012:report/*"
  ]
  ```

  Die in den `aws:SourceArn` UND-Bedingungen angegebenen Konten sollten übereinstimmen. `aws:SourceAccount`

Beide Bedingungen verhindern, dass Amazon Inspector bei Transaktionen mit Amazon S3 als verwirrter Stellvertreter eingesetzt wird. Obwohl wir dies nicht empfehlen, können Sie diese Bedingungen aus der Bucket-Richtlinie entfernen.

9. Wenn Sie mit der Aktualisierung der Bucket-Richtlinie fertig sind, wählen Sie Änderungen speichern aus.

Schritt 3: Konfigurieren Sie eine AWS KMS key

Nachdem Sie Ihre Berechtigungen überprüft und den S3-Bucket konfiguriert haben, legen AWS KMS key Sie fest, welchen Code Amazon Inspector zur Verschlüsselung Ihres Ergebnisberichts verwenden soll. Bei dem Schlüssel muss es sich um einen vom Kunden verwalteten KMS-Schlüssel mit symmetrischer Verschlüsselung handeln. Darüber hinaus muss sich der Schlüssel in demselben befinden AWS-Region wie der S3-Bucket, den Sie zum Speichern des Berichts konfiguriert haben.

Der Schlüssel kann ein vorhandener KMS-Schlüssel aus Ihrem eigenen Konto oder ein vorhandener KMS-Schlüssel sein, der einem anderen Konto gehört. Wenn Sie einen neuen KMS-Schlüssel verwenden möchten, erstellen Sie den Schlüssel, bevor Sie fortfahren. Wenn Sie einen vorhandenen Schlüssel verwenden möchten, der einem anderen Konto gehört, besorgen Sie sich den Amazon-Ressourcennamen (ARN) des Schlüssels. Sie müssen diesen ARN eingeben, wenn Sie Ihren Bericht aus Amazon Inspector exportieren. Informationen zum Erstellen und Überprüfen der Einstellungen für KMS-Schlüssel finden Sie unter Schlüssel verwalten im AWS Key Management Service Entwicklerhandbuch.

Nachdem Sie festgelegt haben, welchen KMS-Schlüssel Sie verwenden möchten, erteilen Sie Amazon Inspector die Erlaubnis, den Schlüssel zu verwenden. Andernfalls kann Amazon Inspector den Bericht nicht verschlüsseln und exportieren. Um Amazon Inspector die Erlaubnis zur Verwendung des Schlüssels zu erteilen, aktualisieren Sie die Schlüsselrichtlinie für den Schlüssel. Detaillierte Informationen zu den wichtigsten Richtlinien und zur Verwaltung des Zugriffs auf KMS-Schlüssel finden Sie AWS KMS im AWS Key Management Service Entwicklerhandbuch unter Wichtige Richtlinien.

> **Note**
>
> Das folgende Verfahren dient zum Aktualisieren eines vorhandenen Schlüssels, damit Amazon Inspector ihn verwenden kann. Wenn Sie noch keinen Schlüssel haben, finden Sie weitere Informationen unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

Um die Schlüsselrichtlinie zu aktualisieren

1. Melden Sie sich mit Ihren Anmeldeinformationen an, und öffnen Sie dann die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
2. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
3. Wählen Sie den KMS-Schlüssel aus, den Sie zum Verschlüsseln des Berichts verwenden möchten. Der Schlüssel muss ein symmetrischer Verschlüsselungsschlüssel (SYMMETRIC_DEFAULT) sein.
4. Wählen Sie im Tab Schlüsselrichtlinie die Option Bearbeiten aus. Wenn Sie keine Schlüsselrichtlinie mit der Schaltfläche Bearbeiten sehen, müssen Sie zuerst Zur Richtlinienansicht wechseln auswählen.
5. Kopieren Sie die folgende Beispielanweisung in Ihre Zwischenablage:

```
{
    "Sid": "Allow Amazon Inspector to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "inspector2.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:inspector2:Region:111122223333:report/*"
        }
    }
}
```

6. Fügen Sie im Schlüsselrichtlinien-Editor auf der AWS KMS Konsole die vorherige Anweisung in die Schlüsselrichtlinie ein, um sie der Richtlinie hinzuzufügen.

   Stellen Sie beim Hinzufügen der Anweisung sicher, dass die Syntax gültig ist. Wichtige Richtlinien verwenden das JSON-Format. Das bedeutet, dass Sie vor oder nach der Anweisung ein Komma hinzufügen müssen, je nachdem, an welcher Stelle Sie die Anweisung zur Richtlinie

hinzufügen. Wenn Sie die Anweisung als letzte Anweisung hinzufügen, fügen Sie nach der schließenden Klammer der vorhergehenden Anweisung ein Komma ein. Wenn Sie die Anweisung als erste Anweisung oder zwischen zwei vorhandenen Anweisungen hinzufügen, fügen Sie nach der schließenden Klammer der Anweisung ein Komma ein.

7. Aktualisieren Sie die Anweisung mit den richtigen Werten für Ihre Umgebung, wobei:

   - *111122223333*ist die Konto-ID für Ihre AWS-Konto.
   - *Region*ist der, AWS-Region in dem Sie Amazon Inspector erlauben möchten, Berichte mit dem Schlüssel zu verschlüsseln. Zum Beispiel `us-east-1` für die Region USA Ost (Nord-Virginia).

   > **Note**
   >
   > Wenn Sie Amazon Inspector in einer manuell aktivierten Umgebung verwenden AWS-Region, fügen Sie dem Wert für das `Service` Feld auch den entsprechenden Regionalcode hinzu. Wenn Sie Amazon Inspector beispielsweise in der Region Naher Osten (Bahrain) verwenden, `inspector2.amazonaws.com` ersetzen Sie durch`inspector2.me-south-1.amazonaws.com`.

   Wie die Beispielanweisung für die Bucket-Richtlinie im vorherigen Schritt verwenden die `Condition` Felder in diesem Beispiel zwei globale IAM-Bedingungsschlüssel:

   - aws: SourceAccount — Diese Bedingung ermöglicht es Amazon Inspector, die angegebenen Aktionen nur für Ihr Konto auszuführen. Insbesondere bestimmt sie, welches Konto die angegebenen Aktionen für die in der `aws:SourceArn` Bedingung angegebenen Ressourcen und Aktionen ausführen kann.

     Damit Amazon Inspector die angegebenen Aktionen für weitere Konten ausführen kann, fügen Sie dieser Bedingung die Konto-ID für jedes weitere Konto hinzu. Beispiel:

     ```
     "aws:SourceAccount": ["111122223333","444455556666","123456789012"]
     ```

   - aws: SourceArn — Diese Bedingung verhindert, dass andere AWS-Services die angegebenen Aktionen ausführen. Es verhindert auch, dass Amazon Inspector den Schlüssel verwendet, während andere Aktionen für Ihr Konto ausgeführt werden. Mit anderen Worten, Amazon Inspector kann S3-Objekte nur dann mit dem Schlüssel verschlüsseln, wenn es sich bei den

Objekten um Ergebnisberichte handelt, und nur, wenn diese Berichte von dem Konto und in der Region erstellt wurden, die in der Bedingung angegeben sind.

Damit Amazon Inspector die angegebenen Aktionen für weitere Konten ausführen kann, fügen Sie dieser Bedingung ARNs für jedes weitere Konto hinzu. Beispiel:

```
"aws:SourceArn": [
    "arn:aws:inspector2:us-east-1:111122223333:report/*",
    "arn:aws:inspector2:us-east-1:444455556666:report/*",
    "arn:aws:inspector2:us-east-1:123456789012:report/*"
]
```

Die in den `aws:SourceArn` UND-Bedingungen angegebenen Konten sollten übereinstimmen. `aws:SourceAccount`

Diese Bedingungen verhindern, dass Amazon Inspector bei Transaktionen mit als verwirrter Stellvertreter eingesetzt wird AWS KMS. Obwohl wir dies nicht empfehlen, können Sie diese Bedingungen aus der Abrechnung entfernen.

8. Wenn Sie mit der Aktualisierung der wichtigsten Richtlinie fertig sind, wählen Sie Änderungen speichern aus.

Schritt 4: Konfigurieren und exportieren Sie einen Ergebnisbericht

> **Note**
>
> Sie können jeweils nur einen Ergebnisbericht exportieren. Wenn gerade ein Export ausgeführt wird, müssen Sie warten, bis der Export abgeschlossen ist, bevor Sie einen weiteren Ergebnisbericht exportieren.

Nachdem Sie Ihre Berechtigungen überprüft und Ressourcen zum Verschlüsseln und Speichern Ihres Ergebnisberichts konfiguriert haben, können Sie den Bericht konfigurieren und exportieren.

Um einen Ergebnisbericht zu konfigurieren und zu exportieren

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.

2. Wählen Sie im Navigationsbereich unter Ergebnisse die Option Alle Ergebnisse aus.
3. (Optional) Fügen Sie mithilfe der Filterleiste über der Tabelle Ergebnisse Filterkriterien hinzu, die angeben, welche Ergebnisse in den Bericht aufgenommen werden sollen. Wenn Sie Kriterien hinzufügen, aktualisiert Amazon Inspector die Tabelle so, dass sie nur die Ergebnisse enthält, die den Kriterien entsprechen. Die Tabelle bietet eine Vorschau der Daten, die Ihr Bericht enthalten wird.

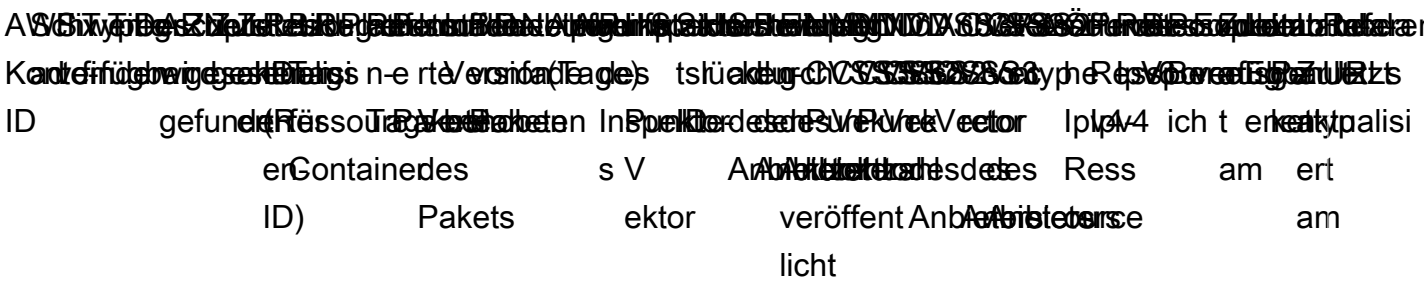
   > **Note**
   >
   > Wir empfehlen, Filterkriterien hinzuzufügen. Wenn Sie dies nicht tun, enthält der Bericht Daten für alle Ihre Ergebnisse in der aktuellen Version AWS-Region , die den Status Aktiv haben. Wenn Sie der Amazon Inspector-Administrator einer Organisation sind, umfasst dies auch Ergebnisdaten für alle Mitgliedskonten in Ihrer Organisation.
   > Wenn ein Bericht Daten für alle oder viele Ergebnisse enthält, kann es lange dauern, den Bericht zu generieren und zu exportieren, und Sie können jeweils nur einen Bericht exportieren.

4. Wählen Sie Ergebnisse exportieren aus.
5. Geben Sie im Abschnitt Exporteinstellungen unter Exportdateityp ein Dateiformat für den Bericht an:

   - Um eine JavaScript Objektnotationsdatei (.json) zu erstellen, die die Daten enthält, wählen Sie JSON.

     Wenn Sie die JSON-Option wählen, enthält der Bericht alle Felder für jedes Ergebnis. Eine Liste möglicher JSON-Felder finden Sie unter dem Datentyp Finding in der Amazon Inspector API-Referenz.
   - Um eine Datei mit kommagetrennten Werten (.csv) zu erstellen, die die Daten enthält, wählen Sie CSV.

     Wenn Sie die CSV-Option wählen, enthält der Bericht nur eine Teilmenge der Felder für jedes Ergebnis, also etwa 45 Felder, die die wichtigsten Attribute eines Ergebnisses angeben. Zu den Feldern gehören: Art des Ergebnisses, Titel, Schweregrad, Status, Beschreibung, Zuerst gesehen, Zuletzt gesehen, Fix verfügbar, AWS Konto-ID, Ressourcen-ID, Ressourcen-Tags und Korrektur. Diese Felder ergänzen die Felder, in denen Bewertungsdetails und Referenz-URLs für jedes Ergebnis erfasst werden. Im Folgenden finden Sie ein Beispiel für die CSV-Header in einem Ergebnisbericht:



6. Geben Sie unter Exportspeicherort für S3-URI den S3-Bucket an, in dem Sie den Bericht speichern möchten:

   - Um den Bericht in einem Bucket zu speichern, der Ihrem Konto gehört, wählen Sie Browse S3 aus. Amazon Inspector zeigt eine Tabelle der S3-Buckets für Ihr Konto an. Wählen Sie die Zeile für den gewünschten Bucket aus, und klicken Sie dann auf Choose.

     > **Tip**
     >
     > Um auch ein Amazon S3-Pfadpräfix für den Bericht anzugeben, fügen Sie einen Schrägstrich (/) und das Präfix an den Wert im S3-URI-Feld an. Amazon Inspector bezieht dann das Präfix ein, wenn es den Bericht zum Bucket hinzufügt, und Amazon S3 generiert den durch das Präfix angegebenen Pfad.
     > Wenn Sie beispielsweise Ihre AWS-Konto ID als Präfix verwenden möchten und Ihre Konto-ID 111122223333 lautet, fügen Sie den Wert im Feld **`/111122223333`** S3-URI an.
     > Ein Präfix ähnelt einem Verzeichnispfad in einem S3-Bucket. Es ermöglicht Ihnen, ähnliche Objekte in einem Bucket zu gruppieren, ähnlich wie Sie ähnliche Dateien zusammen in einem Ordner auf einem Dateisystem speichern könnten. Weitere Informationen finden Sie unter Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern im Amazon Simple Storage Service-Benutzerhandbuch.

   - Um den Bericht in einem Bucket zu speichern, das einem anderen Konto gehört, geben Sie den URI für den Bucket ein — zum Beispiel**`s3://DOC-EXAMPLE_BUCKET`**, wobei DOC-EXAMPLE _BUCKET der Name des Buckets ist. Der Bucket-Besitzer kann diese Informationen für Sie in den Eigenschaften des Buckets finden.

7. Geben Sie für den KMS-Schlüssel den an AWS KMS key , den Sie zum Verschlüsseln des Berichts verwenden möchten:

- Um einen Schlüssel aus Ihrem eigenen Konto zu verwenden, wählen Sie den Schlüssel aus der Liste aus. In der Liste werden vom Kunden verwaltete KMS-Schlüssel mit symmetrischer Verschlüsselung für Ihr Konto angezeigt.
- Um einen Schlüssel zu verwenden, der einem anderen Konto gehört, geben Sie den Amazon-Ressourcennamen (ARN) des Schlüssels ein. Der Schlüsselbesitzer kann diese Informationen für Sie in den Eigenschaften des Schlüssels finden. Weitere Informationen finden Sie im AWS Key Management Service Entwicklerhandbuch unter Suchen der Schlüssel-ID und des Schlüssel-ARN.

8. Wählen Sie Export aus.

Amazon Inspector generiert den Ergebnisbericht, verschlüsselt ihn mit dem von Ihnen angegebenen KMS-Schlüssel und fügt ihn dem von Ihnen angegebenen S3-Bucket hinzu. Abhängig von der Anzahl der Ergebnisse, die Sie in den Bericht aufnehmen möchten, kann dieser Vorgang mehrere Minuten oder Stunden dauern. Wenn der Export abgeschlossen ist, zeigt Amazon Inspector eine Meldung an, dass Ihr Ergebnisbericht erfolgreich exportiert wurde. Wählen Sie optional Bericht anzeigen in der Nachricht aus, um zum Bericht in Amazon S3 zu navigieren.

> **Note**
>
> Bei der Berichtsgenerierung kommt es zu einem Timeout, wenn die Fertigstellung länger als 12 Stunden dauert. Wenn bei Ihrem Bericht ein Timeout auftritt, fügen Sie Filterkriterien hinzu, um die Anzahl der im Bericht enthaltenen Ergebnisse zu reduzieren, und exportieren Sie den Bericht dann erneut.

Beachten Sie, dass Sie jeweils nur einen Bericht exportieren können. Wenn gerade ein Export ausgeführt wird, warten Sie, bis dieser Export abgeschlossen ist, bevor Sie versuchen, einen weiteren Bericht zu exportieren.

Beheben Sie Exportfehler

Wenn beim Versuch, einen Ergebnisbericht zu exportieren, ein Fehler auftritt, zeigt Amazon Inspector eine Meldung mit einer Beschreibung des Fehlers an. Sie können die Informationen in diesem Thema als Leitfaden verwenden, um mögliche Ursachen und Lösungen für den Fehler zu identifizieren.

Stellen Sie beispielsweise sicher, dass sich der S3-Bucket im aktuellen Bucket befindet AWS-Region und die Richtlinie des Buckets Amazon Inspector das Hinzufügen von Objekten zum Bucket

ermöglicht. Stellen Sie außerdem sicher, dass der in der aktuellen Region aktiviert AWS KMS key ist, und stellen Sie sicher, dass Amazon Inspector den Schlüssel gemäß der Schlüsselrichtlinie verwenden kann.

Nachdem Sie den Fehler behoben haben, versuchen Sie erneut, den Bericht zu exportieren.

Fehler: Es können nicht mehrere Berichte vorliegen

Wenn Sie versuchen, einen Bericht zu erstellen, Amazon Inspector jedoch bereits einen Bericht generiert, erhalten Sie eine Fehlermeldung mit der Angabe Grund: Es können nicht mehrere Berichte in Bearbeitung sein. Dieser Fehler tritt auf, weil Amazon Inspector jeweils nur einen Bericht für ein Konto erstellen kann.

Um den Fehler zu beheben, können Sie warten, bis der andere Bericht abgeschlossen ist, oder ihn stornieren, bevor Sie einen neuen Bericht anfordern.

Sie können den Status eines Berichts überprüfen, indem Sie den GetFindingsReportStatus Vorgang verwenden. Dieser Vorgang gibt die Berichts-ID jedes Berichts zurück, der gerade generiert wird.

Bei Bedarf können Sie die durch den `GetFindingsReportStatus` Vorgang angegebene Berichts-ID verwenden, um einen gerade ausgeführten Export mithilfe des CancelFindingsReport Vorgangs abzubrechen.

Erstellung benutzerdefinierter Antworten auf Amazon Inspector-Ergebnisse mit Amazon EventBridge

Amazon Inspector erstellt in Amazon ein Ereignis EventBridge für neu generierte und aggregierte Ergebnisse. Amazon Inspector erstellt außerdem ein Ereignis für alle Änderungen am Status eines Ergebnisses. Das bedeutet, dass Amazon Inspector ein neues Ereignis für einen Befund erstellt, wenn Sie Maßnahmen wie den Neustart einer Ressource oder das Ändern der mit einer Ressource verknüpften Tags ergreifen. Wenn Amazon Inspector ein neues Ereignis für ein aktualisiertes Ergebnis erstellt, `id` bleibt das Ergebnis unverändert.

> **Note**
>
> Wenn es sich bei Ihrem Konto um ein von Amazon Inspector delegiertes Administratorkonto handelt, werden Ereignisse auf Ihrem Konto und dem Mitgliedskonto EventBridge veröffentlicht, von dem die Ereignisse ausgegangen sind.

Wenn Sie EventBridge Ereignisse mit Amazon Inspector verwenden, können Sie Aufgaben automatisieren, um auf Sicherheitsprobleme zu reagieren, die Ihre Ergebnisse aufdecken. Um Benachrichtigungen über Ergebnisse von Amazon Inspector auf der Grundlage von EventBridge Ereignissen zu erhalten, müssen Sie eine EventBridge Regel erstellen und ein Ziel für Amazon Inspector angeben. Die EventBridge Regel EventBridge ermöglicht das Senden von Benachrichtigungen für Amazon Inspector-Ergebnisse, und das Ziel gibt an, wohin die Benachrichtigungen gesendet werden sollen.

Amazon Inspector sendet Ereignisse an den Standard-Event-Bus an AWS-Region dem Ort, an dem Sie Amazon Inspector gerade verwenden. Das bedeutet, dass Sie für jeden AWS-Region Ort, an dem Sie Amazon Inspector aktiviert und Amazon Inspector für den Empfang von EventBridge Ereignissen konfiguriert haben, Ereignisregeln konfigurieren müssen. Amazon Inspector sendet Ereignisse nach bestem Bemühen aus.

In diesem Abschnitt finden Sie ein Beispiel für ein Eventschema und es wird beschrieben, wie Sie eine EventBridge Regel erstellen.

Ereignisschema

Das Folgende ist ein Beispiel für das Amazon Inspector-Ereignisformat für ein EC2-Findereignis. Ein Beispiel für ein Schema anderer Ergebnisarten und Ereignistypen finden Sie unterEventBridge Schema.

```
{
    "version": "0",
    "id": "66a7a279-5f92-971c-6d3e-c92da0950992",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-19T22:46:15Z",
    "region": "us-east-1",
    "resources": ["i-0c2a343f1948d5205"],
    "detail": {
        "awsAccountId": "111122223333",
        "description": "\n It was discovered that the sound subsystem in the Linux
 kernel contained a\n race condition in some situations. A local attacker could use
 this to cause\n a denial of service (system crash).",
        "exploitAvailable": "YES",
        "exploitabilityDetails": {
            "lastKnownExploitAt": "Oct 24, 2022, 11:08:59 PM"
```

```
        },
        "findingArn": "arn:aws:inspector2:us-east-1:111122223333:finding/FINDING_ID",
        "firstObservedAt": "Jan 19, 2023, 10:46:15 PM",
        "fixAvailable": "YES",
        "lastObservedAt": "Jan 19, 2023, 10:46:15 PM",
        "packageVulnerabilityDetails": {
            "cvss": [{
                "baseScore": 4.7,
                "scoringVector": "CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H",
                "source": "NVD",
                "version": "3.1"
            }],
            "referenceUrls": ["https://lore.kernel.org/all/
CAFcO6XN7JDM4xSXGhtusQfS2mSBcx50VJKwQpCq=WeLt57aaZA@mail.gmail.com/", "https://
ubuntu.com/security/notices/USN-5792-1", "https://ubuntu.com/security/notices/
USN-5791-2", "https://ubuntu.com/security/notices/USN-5791-1", "https://ubuntu.com/
security/notices/USN-5793-2", "https://git.kernel.org/pub/scm/linux/kernel/git/
torvalds/linux.git/commit/?id=8423f0b6d513b259fdab9c9bf4aaa6188d054c2d", "https://
ubuntu.com/security/notices/USN-5793-1", "https://ubuntu.com/security/notices/
USN-5792-2", "https://ubuntu.com/security/notices/USN-5791-3", "https://ubuntu.com/
security/notices/USN-5793-4", "https://ubuntu.com/security/notices/USN-5793-3",
 "https://git.kernel.org/linus/8423f0b6d513b259fdab9c9bf4aaa6188d054c2d(6.0-rc5)",
 "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3303"],
            "relatedVulnerabilities": [],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2022/
CVE-2022-3303.html",
            "vendorCreatedAt": "Sep 27, 2022, 11:15:00 PM",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2022-3303",
            "vulnerablePackages": [{
                "arch": "X86_64",
                "epoch": 0,
                "fixedInVersion": "0:5.15.0.1027.31~20.04.16",
                "name": "linux-image-aws",
                "packageManager": "OS",
                "remediation": "apt update && apt install --only-upgrade linux-image-
aws",
                "version": "5.15.0.1026.30~20.04.16"
            }]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
```

```
            }
        },
        "resources": [{
            "details": {
                "awsEc2Instance": {
                    "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-
profile/AmazonSSMRoleForInstancesQuickSetup",
                    "imageId": "ami-0b7ff1a8d69f1bb35",
                    "ipV4Addresses": ["172.31.85.212", "44.203.45.27"],
                    "ipV6Addresses": [],
                    "launchedAt": "Jan 19, 2023, 7:53:14 PM",
                    "platform": "UBUNTU_20_04",
                    "subnetId": "subnet-8213f2a3",
                    "type": "t2.micro",
                    "vpcId": "vpc-ab6650d1"
                }
            },
            "id": "i-0c2a343f1948d5205",
            "partition": "aws",
            "region": "us-east-1",
            "type": "AWS_EC2_INSTANCE"
        }],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "CVE-2022-3303 - linux-image-aws",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Jan 19, 2023, 10:46:15 PM"
    }
}
```

Erstellung einer EventBridge Regel, um Sie über Ergebnisse von Amazon Inspector zu informieren

Um die Sichtbarkeit der Ergebnisse von Amazon Inspector EventBridge zu erhöhen, können Sie automatische Warnmeldungen einrichten, die an einen Messaging-Hub gesendet werden. In diesem Thema erfahren Sie, wie Sie Warnmeldungen `CRITICAL` und Hinweise zum `HIGH` Schweregrad per E-Mail, Slack oder Amazon Chime senden. Sie erfahren, wie Sie ein Amazon Simple Notification Service-Thema einrichten und dieses Thema dann mit einer EventBridge Event-Regel verbinden.

Schritt 1. Richten Sie ein Amazon SNS-Thema und einen Endpunkt ein

Um automatische Benachrichtigungen einzurichten, müssen Sie zunächst ein Thema in Amazon Simple Notification Service einrichten und einen Endpunkt hinzufügen. Weitere Informationen finden Sie im SNS-Handbuch.

Dieses Verfahren legt fest, wohin Sie die Daten zu den Ergebnissen von Amazon Inspector senden möchten. Das SNS-Thema kann während oder nach der Erstellung der EventBridge Ereignisregel zu einer Ereignisregel hinzugefügt werden.

Email setup

Erstellen eines SNS-Themas

1. Melden Sie sich bei der Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home an.
2. Wählen Sie im Navigationsbereich Themen und dann Thema erstellen aus.
3. Wählen Sie im Abschnitt Thema erstellen die Option Standard aus. Geben Sie als Nächstes einen Themennamen ein, z. **`Inspector_to_Email`** B. Weitere Angaben sind optional.
4. Wählen Sie Create Topic (Thema erstellen) aus. Dadurch wird ein neues Fenster mit Details zu Ihrem neuen Thema geöffnet.
5. Wählen Sie im Abschnitt Abonnements die Option Abonnement erstellen aus.
6. a. Wählen Sie im Menü Protocol (Protokoll) die Option Email (E-Mail) aus.
   b. Geben Sie im Feld Endpunkt die E-Mail-Adresse ein, an die Sie Benachrichtigungen erhalten möchten.

      > **Note**
      >
      > Nachdem Sie das Abonnement erstellt haben, müssen Sie Ihr Abonnement über Ihren E-Mail-Client bestätigen.

   c. Wählen Sie Create subscription (Abonnement erstellen) aus.
7. Suchen Sie in Ihrem Posteingang nach einer Abonnementnachricht und wählen Sie Abonnement bestätigen.

Slack setup

Erstellen eines SNS-Themas

1. Melden Sie sich bei der Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home an.
2. Wählen Sie im Navigationsbereich Themen und dann Thema erstellen aus.
3. Wählen Sie im Abschnitt Thema erstellen die Option Standard aus. Geben Sie als Nächstes einen Themennamen ein, z. **`Inspector_to_Slack`** B. Weitere Angaben sind optional. Wählen Sie Thema erstellen, um die Endpunkterstellung abzuschließen.

Konfiguration eines Amazon Q Developer im Client für Chat-Anwendungen

1. Navigieren Sie in der Konsole für Chat-Anwendungen unter zum Amazon Q Developerhttps://console.aws.amazon.com/chatbot/.
2. Wählen Sie im Bereich Konfigurierte Kunden die Option Neuen Client konfigurieren aus.
3. Wähle Slack und dann zur Bestätigung „Konfigurieren“.

> **Note**
>
> Wenn du dich für Slack entscheidest, musst du in Chat-Anwendungen die Berechtigungen für Amazon Q Developer für den Zugriff auf deinen Kanal bestätigen, indem du Zulassen auswählst.

4. Wählen Sie Neuen Kanal konfigurieren aus, um den Bereich mit den Konfigurationsdetails zu öffnen.
   a. Geben Sie einen Namen für den Kanal ein.
   b. Wählen Sie für den Slack-Kanal den Kanal aus, den Sie verwenden möchten.
   c. Kopiere in Slack die Kanal-ID des privaten Channels, indem du mit der rechten Maustaste auf den Kanalnamen klickst und Link kopieren auswählst.
   d. Fügen Sie im Fenster Amazon Q Developer in Chat-Anwendungen die Kanal-ID, die Sie aus Slack kopiert haben, in das Feld Private Channel-ID ein. AWS-Managementkonsole
   e. Wähle unter Berechtigungen aus, ob du mithilfe einer Vorlage eine IAM-Rolle erstellen möchtest, falls du noch keine Rolle hast.

f. Wählen Sie für Richtlinienvorlagen die Option Benachrichtigungsberechtigungen aus. Dies ist die IAM-Richtlinienvorlage für Amazon Q Developer in Chat-Anwendungen. Diese Richtlinie bietet die erforderlichen Lese- und Listenberechtigungen für CloudWatch Alarme, Ereignisse und Protokolle sowie für Amazon SNS-Themen.

g. Für Channel-Guardrail-Richtlinien wählen Sie. AmazonInspector2ReadOnlyAccess

h. Wählen Sie die Region aus, in der Sie zuvor Ihr SNS-Thema erstellt haben, und wählen Sie dann das Amazon SNS-Thema aus, das Sie erstellt haben, um Benachrichtigungen an den Slack-Kanal zu senden.

5. Wählen Sie Konfigurieren.

Amazon Chime setup

Erstellen eines SNS-Themas

1. Melden Sie sich bei der Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home an.
2. Wählen Sie im Navigationsbereich Themen aus und wählen Sie dann Thema erstellen aus.
3. Wählen Sie im Abschnitt Thema erstellen die Option Standard aus. Geben Sie als Nächstes einen Themennamen ein, z. **`Inspector_to_Chime`** B. Weitere Angaben sind optional. Wählen Sie „Thema erstellen“, um den Vorgang abzuschließen.

Konfiguration eines Amazon Q Developer im Client für Chat-Anwendungen

1. Navigieren Sie in der Konsole für Chat-Anwendungen unter zum Amazon Q Developerhttps://console.aws.amazon.com/chatbot/.
2. Wählen Sie im Bereich Konfigurierte Clients die Option Neuen Client konfigurieren.
3. Wählen Sie Chime und dann zur Bestätigung Configure.
4. Geben Sie im Bereich mit den Konfigurationsdetails einen Namen für den Kanal ein.
5. Öffnen Sie in Amazon Chime den gewünschten Chatraum.

   a. Wählen Sie das Zahnradsymbol rechts oben und danach Manage webhooks and bots aus.

   b. Wählen Sie URL kopieren, um die Webhook-URL in Ihre Zwischenablage zu kopieren.

6. Fügen Sie im Fenster Amazon Q Developer in Chat-Anwendungen die URL, die Sie kopiert haben, in das Feld Webhook-URL ein. AWS-Managementkonsole

7. Wählen Sie unter Berechtigungen aus, ob Sie mithilfe einer Vorlage eine IAM-Rolle erstellen möchten, falls Sie noch keine Rolle haben.
8. Wählen Sie für Richtlinienvorlagen die Option Benachrichtigungsberechtigungen aus. Dies ist die IAM-Richtlinienvorlage für Amazon Q Developer in Chat-Anwendungen. Sie bietet die erforderlichen Lese- und Listenberechtigungen für CloudWatch Alarme, Ereignisse und Protokolle sowie für Amazon SNS-Themen.
9. Wählen Sie die Region aus, in der Sie zuvor Ihr SNS-Thema erstellt haben, und wählen Sie dann das Amazon SNS-Thema aus, das Sie erstellt haben, um Benachrichtigungen an den Amazon Chime-Raum zu senden.
10. Wählen Sie Konfigurieren.

Schritt 2. Erstellen Sie eine EventBridge Regel für die Ergebnisse von Amazon Inspector

1. Melden Sie sich mit Ihren Zugangsdaten an.
2. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
3. Wählen Sie im Navigationsbereich Regeln aus und wählen Sie dann Regel erstellen aus.
4. Geben Sie einen Namen und optional eine Beschreibung für Ihre Regel ein.
5. Wählen Sie Regel mit einem Ereignismuster und dann Weiter aus.
6. Wählen Sie im Bereich Event-Muster die Option Benutzerdefinierte Muster (JSON-Editor) aus.
7. Fügen Sie den folgenden JSON-Code in den Editor ein.

```
{
  "source": ["aws.inspector2"],
  "detail-type": ["Inspector2 Finding"],
  "detail": {
    "severity": ["HIGH", "CRITICAL"],
    "status": ["ACTIVE"]
  }
}
```

> **Note**
>
> Dieses Muster sendet Benachrichtigungen über alle aktiven `CRITICAL` oder schwerwiegenden Befunde`HIGH`, die von Amazon Inspector erkannt wurden.

Wählen Sie Weiter, wenn Sie mit der Eingabe des Ereignismusters fertig sind.

8. Wählen Sie auf der Seite „Ziele auswählen“ AWS-Service. Wählen Sie dann für Zieltyp auswählen das SNS-Thema aus.
9. Wählen Sie für Thema den Namen des SNS-Themas aus, das Sie in Schritt 1 erstellt haben. Klicken Sie anschließend auf Weiter.
10. Fügen Sie bei Bedarf optionale Tags hinzu und wählen Sie Weiter.
11. Überprüfen Sie Ihre Regel und wählen Sie dann Regel erstellen aus.

EventBridge für Amazon Inspector-Umgebungen mit mehreren Konten

Wenn Sie ein von Amazon Inspector beauftragter Administrator sind, werden in Ihrem Konto EventBridge Regeln angezeigt, die auf den entsprechenden Ergebnissen Ihrer Mitgliedskonten basieren. Wenn Sie EventBridge in Ihrem Administratorkonto Benachrichtigungen über Ergebnisse einrichten, wie im vorherigen Abschnitt beschrieben, erhalten Sie Benachrichtigungen über mehrere Konten. Mit anderen Worten, Sie werden über Ergebnisse und Ereignisse informiert, die von Ihren Mitgliedskonten generiert wurden, zusätzlich zu denen, die von Ihrem eigenen Konto generiert wurden.

Sie können anhand der `accountId` JSON-Details des Ergebnisses das Mitgliedskonto identifizieren, von dem das Amazon Inspector-Ergebnis stammt.

Arbeiten mit dem Dashboard in Amazon Inspector

Das Dashboard bietet eine Momentaufnahme der aggregierten Statistiken für Ressourcen, die Amazon Inspector scannt. Verwenden Sie das Dashboard, um mehr über die Abdeckung Ihrer Umgebung und wichtige Ergebnisse zu erfahren.

> **Note**
>
> Wenn es sich bei Ihrem Konto um das delegierte Administratorkonto für eine Organisation handelt, werden im Dashboard Informationen zu Ihrem Konto und allen anderen Konten in der Organisation angezeigt.

In diesem Thema wird beschrieben, wie Sie das Dashboard anzeigen und die Komponenten verstehen, aus denen das Dashboard besteht.

Themen

- Das Dashboard anzeigen
- Dashboard-Komponenten verstehen und Daten interpretieren

Das Dashboard anzeigen

Das Dashboard bietet einen Überblick über den Versicherungsschutz für Ihre Umgebung und wichtige Ergebnisse. Das Dashboard aktualisiert die Daten automatisch alle fünf Minuten. Sie können Daten manuell aktualisieren, indem Sie das Aktualisierungssymbol in der oberen rechten Ecke des Bildschirms auswählen. Sie können unterstützende Daten zu einem Artikel anzeigen, indem Sie den Artikel auswählen.

> **Note**
>
> Wenn es sich bei Ihrem Konto um das delegierte Administratorkonto für eine Organisation handelt, können Sie aggregierte Statistiken für ein Mitgliedskonto anzeigen, indem Sie die Mitgliedskonto-ID in das Feld Konto eingeben.

So zeigen Sie das Dashboard an:

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Dashboard aus.

Dashboard-Komponenten verstehen und Daten interpretieren

Jeder Abschnitt des Dashboards bietet Einblicke in wichtige Kennzahlen und Ergebnisdaten, sodass Sie sich ein Bild von der aktuellen Sicherheitsanfälligkeit Ihrer AWS Ressourcen machen können AWS-Region.

Abdeckung der Umwelt

Der Abschnitt Umweltberichterstattung enthält Statistiken über die von Amazon Inspector gescannten Ressourcen. In diesem Abschnitt können Sie die Anzahl und den Prozentsatz der EC2 Amazon-Instances, Amazon ECR-Bilder und AWS Lambda Funktionen sehen, die von Amazon Inspector gescannt wurden. Wenn Sie AWS Organizations als delegierter Administrator von Amazon Inspector mehrere Konten verwalten, werden Ihnen auch die Gesamtzahl der Organisationskonten, die Anzahl mit aktiviertem Amazon Inspector und der daraus resultierende Deckungsgrad für die Organisation angezeigt. In diesem Abschnitt können Sie auch feststellen, welche Ressourcen nicht von Amazon Inspector abgedeckt werden. Diese Ressourcen können Sicherheitslücken enthalten, die ausgenutzt werden könnten, um Ihr Unternehmen zu gefährden. Weitere Details finden Sie unter Bewertung der Abdeckung Ihrer AWS Umgebung durch Amazon Inspector.

Wenn Sie eine Deckungsgruppe auswählen, gelangen Sie zur Kontoverwaltungsseite für die von Ihnen gewählte Gruppierung. Auf der Kontoverwaltungsseite finden Sie Details darüber, welche Konten, EC2 Amazon-Instances und Amazon ECR-Repositorys von Amazon Inspector abgedeckt werden.

Die folgenden Deckungsgruppen sind verfügbar:

- Account
- Instances
- Container-Repositorien
- Container-Images
- Lambda

Kritische Ergebnisse

Der Abschnitt Kritische Ergebnisse enthält eine Anzahl der kritischen Sicherheitslücken in Ihrer Umgebung und eine Gesamtzahl aller Ergebnisse in Ihrer Umgebung. In diesem Abschnitt werden die Zahlen pro Ressource und Bewertungstyp angezeigt. Weitere Informationen zu kritischen Ergebnissen und dazu, wie Amazon Inspector die Kritikalität bestimmt, finden Sie unterDie Ergebnisse von Amazon Inspector verstehen.

Wenn Sie eine kritische Ergebnisgruppe auswählen, gelangen Sie zur Seite Alle Ergebnisse und wendet automatisch Filter an, um alle kritischen Ergebnisse anzuzeigen, die der von Ihnen ausgewählten Gruppierung entsprechen.

Die folgenden Gruppen kritischer Ergebnisse sind verfügbar:

- Ergebnisse des Amazon Inspector Inspector-Codescans
- Ergebnisse der EC2 Amazon-Instance
- Ergebnisse des Amazon ECR-Container-Images
- Ergebnisse der Lambda-Funktion

Risikobasierte Abhilfemaßnahmen

Im Abschnitt Risikobasierte Problembehebungen werden die fünf Softwarepakete mit den wichtigsten Sicherheitslücken aufgeführt, von denen die meisten Ressourcen in Ihrer Umgebung betroffen sind. Durch die Behebung dieser Pakete kann die Anzahl kritischer Risiken für Ihre Umgebung erheblich reduziert werden. Wählen Sie den Namen des Softwarepakets, um die zugehörigen Sicherheitslücken und die betroffenen Ressourcen zu sehen.

Konten mit den wichtigsten Ergebnissen

Im Abschnitt Konten mit den kritischsten Ergebnissen werden die fünf AWS Konten in Ihrer Umgebung mit den kritischsten Ergebnissen sowie die Gesamtzahl der Ergebnisse für dieses Konto angezeigt. Dieser Abschnitt ist nur vom delegierten Administratorkonto aus sichtbar, wenn Amazon Inspector für das Scannen mehrerer Konten mit konfiguriert ist. AWS Organizations Diese Ansicht hilft delegierten Administratoren zu verstehen, welche Konten innerhalb des Unternehmens möglicherweise am stärksten gefährdet sind.

Wählen Sie Konto-ID, um weitere Informationen über das betroffene Mitgliedskonto zu erhalten.

Amazon ECR-Repositorys mit den wichtigsten Ergebnissen

Im Abschnitt Elastic Container Registry (ECR) -Repositorien mit den wichtigsten Ergebnissen werden die fünf Amazon ECR-Repositorys in Ihrer Umgebung mit den kritischsten Container-

Image-Ergebnissen angezeigt. In der Ansicht werden der Repository-Name, die AWS Konto-ID, das Erstellungsdatum des Repositorys, die Anzahl der kritischen Sicherheitslücken und die Gesamtzahl der Sicherheitslücken angezeigt. Anhand dieser Ansicht können Sie ermitteln, welche Repositorys möglicherweise am stärksten gefährdet sind.

Wählen Sie den Repository-Namen, um weitere Informationen über das betroffene Repository zu erhalten.

Container-Images mit den wichtigsten Ergebnissen

Im Abschnitt Container-Images mit den kritischsten Ergebnissen werden die fünf Container-Images in Ihrer Umgebung mit den kritischsten Ergebnissen angezeigt. In der Ansicht werden Image-Tag-Daten, Repository-Name, Image-Digest, AWS Konto-ID, Anzahl kritischer Sicherheitslücken und Gesamtzahl der Sicherheitslücken angezeigt. Anhand dieser Ansicht können Anwendungsbesitzer erkennen, welche Container-Images möglicherweise neu erstellt und neu gestartet werden müssen.

Wählen Sie Container-Image, um weitere Informationen über das betroffene Container-Image zu erhalten.

Fälle mit den kritischsten Ergebnissen

Der Abschnitt Instances mit den kritischsten Ergebnissen zeigt die fünf EC2 Amazon-Instances mit den kritischsten Ergebnissen. Die Ansicht zeigt die Instanz-ID, die AWS Konto-ID, die Amazon Machine Image (AMI) -ID, die Anzahl kritischer Sicherheitslücken und die Gesamtzahl der Sicherheitslücken. Diese Ansicht hilft Infrastrukturbesitzern dabei, zu erkennen, welche Instances möglicherweise gepatcht werden müssen.

Wählen Sie Instance-ID, um weitere Informationen über die betroffene EC2 Amazon-Instance zu erhalten.

Amazon Machine Images (AMI) mit den wichtigsten Ergebnissen

Im Abschnitt Amazon Machine Images (AMIs) mit den kritischsten Ergebnissen werden die fünf wichtigsten Ergebnisse AMIs in Ihrer Umgebung angezeigt. Die Ansicht zeigt die AMI-ID, die AWS Konto-ID, die Anzahl der betroffenen EC2 Instances, die in der Umgebung ausgeführt werden, das AMI-Erstellungsdatum, die Betriebssystemplattform des AMI, die Anzahl der kritischen Sicherheitslücken und die Gesamtzahl der Sicherheitslücken. Diese Ansicht hilft Infrastrukturbesitzern dabei, herauszufinden, welche Infrastruktur AMIs möglicherweise neu aufgebaut werden muss.

Wählen Sie Betroffene Instances aus, um weitere Informationen zu den Instances zu erhalten, die über das betroffene AMI gestartet wurden.

AWS Lambda Funktionen mit den wichtigsten Ergebnissen

Im Abschnitt AWS Lambda Funktionen mit den kritischsten Ergebnissen werden die fünf wichtigsten Lambda-Funktionen in Ihrer Umgebung mit den kritischsten Ergebnissen angezeigt. Die Ansicht zeigt den Namen der Lambda-Funktion, die AWS Konto-ID, die Laufzeitumgebung, die Anzahl kritischer Sicherheitslücken, die Anzahl der schwerwiegenden Sicherheitslücken und die Gesamtzahl der Sicherheitslücken. Diese Ansicht hilft Infrastrukturbesitzern dabei, zu erkennen, welche Lambda-Funktionen möglicherweise behoben werden müssen.

Wählen Sie Funktionsname, um weitere Informationen über die betroffene AWS Lambda Funktion zu erhalten.

Amazon Inspector Inspector-Codescans mit den wichtigsten Ergebnissen

Der Abschnitt Projekte mit den kritischsten Code-Schwachstellen zeigt die fünf Projekte mit den wichtigsten Ergebnissen. Sie können ein Projekt auswählen, um Details zu den Ergebnissen anzuzeigen. Wenn Sie ein Projekt auswählen, werden Sie zu dem Repository weitergeleitet, in dem sich die Ergebnisse befinden. Auf der Registerkarte „Ergebnisse“ werden die Namen Ihrer Ergebnisse und deren Schweregrad angezeigt. Es zeigt, welche Art von Analyse zur Generierung Ihrer Ergebnisse verwendet wurde. Es zeigt auch, wie alt Ihre Ergebnisse sind und welchen Status sie haben.

Durchsuchen der Amazon Inspector Inspector-Schwachstellendatenbank

Sie können die Sicherheitslückendatenbank von Amazon Inspector nach häufigen Sicherheitslücken und Risiken (CVE) durchsuchen. Amazon Inspector verwendet Informationen aus der Schwachstellendatenbank, um Details zu einer CVE-ID zu erstellen. Sie können diese Details auf dem Bildschirm mit den CVE-Details einsehen. Amazon Inspector verfolgt und erstellt Ergebnisse für Softwareschwachstellen in der Schwachstellendatenbank. Amazon Inspector unterstützt CVEs nur Plattformen, die im Abschnitt Erkennungsplattformen des CVE-Detailbildschirms aufgeführt sind. In diesem Abschnitt wird beschrieben, wie Sie die Amazon Inspector Inspector-Schwachstellendatenbank mithilfe einer CVE-ID durchsuchen.

> **Note**
>
> Derzeit wird die CVE-Suche nicht unterstützt. Microsoft Windows

Die Schwachstellen-Datenbank durchsuchen

In diesem Abschnitt wird beschrieben, wie Sie die Schwachstellendatenbank in der Konsole und mit der Amazon Inspector API durchsuchen.

> **Note**
>
> Sie müssen Amazon Inspector in Ihrem aktuellen System aktivieren, AWS-Region bevor Sie die Schwachstellendatenbank durchsuchen können.

Console

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home
2. Wählen Sie im Navigationsbereich die Option Vulnerability database search aus.
3. Geben Sie in der Suchleiste eine CVE-ID ein und wählen Sie Suchen aus.

API

Führen Sie die Amazon Inspector SearchVulnerabilitiesInspector-API aus und geben Sie eine einzelne CVE-ID wie `filterCriteria` im folgenden Format ein:`CVE-<year>-<ID>`.

CVE-Details verstehen

In diesem Abschnitt wird beschrieben, wie die CVE-Detailseite zu interpretieren ist.

CVE-Details

Der Abschnitt mit den CVE-Details enthält die folgenden Informationen:

- Beschreibung und ID des CVE
- CVE-Schweregrad
- Bewertungen des Common Vulnerability Scoring System (CVSS) und des Exploit Prediction Scoring System (EPSS)
- Plattformen zur Erkennung

> **Note**
>
> Wenn dieses Feld leer ist, unterstützt Amazon Inspector keine Erkennung für Ihre CVE-ID.

- Aufzählung der häufigsten Schwächen (CWE)
- Datum der Erstellung und Aktualisierung durch den Anbieter

Informationen zu Sicherheitslücken

Der Bereich Vulnerability Intelligence bietet Bedrohungsdaten wie Exploit-Ziele und das Datum des letzten bekannten öffentlichen Exploits.

Es enthält auch Daten der Cybersecurity and Infrastructure Security Agency (CISA), darunter die Abhilfemaßnahmen, das Datum, an dem das CVE in den Katalog der bekannten Sicherheitslücken aufgenommen wurde, und das Datum, zu dem die CISA erwartet, dass Bundesbehörden das CVE beheben werden.

Referenzen

Der Abschnitt „Referenzen“ enthält Links zu Ressourcen mit weiteren Informationen über das CVE.

Exportieren SBOMs mit Amazon Inspector

Eine Software-Stückliste (SBOM) ist ein verschachteltes Inventar aller Open-Source-Softwarekomponenten und Softwarekomponenten von Drittanbietern in Ihrer Codebasis. Amazon Inspector stellt SBOMs einzelne Ressourcen in Ihrer Umgebung bereit. Sie können die Amazon Inspector Inspector-Konsole oder die Amazon Inspector Inspector-API verwenden, um SBOMs für Ihre Ressourcen zu generieren. Sie können SBOMs für alle Ressourcen exportieren, die Amazon Inspector unterstützt und überwacht. Exportiert SBOMs bieten Informationen zu Ihrem Softwareangebot. Sie können den Status Ihrer Ressourcen überprüfen, indem Sie die Abdeckung Ihrer AWS Umgebung bewerten. In diesem Abschnitt wird die Konfiguration und der Export beschrieben SBOMs.

Einige Softwarekomponenten und Paketmanager verwenden Versionsbereiche oder dynamische Verweise anstelle von festen Versionen für Abhängigkeiten. Diese Vorgehensweise erzeugt ungelöste Hashes, bei denen Amazon Inspector eine Hash- oder JAR-Datei identifiziert, sie aber nicht einem bestimmten Namen und einer bestimmten Version zur Erkennung von Sicherheitslücken zuordnen kann. Amazon Inspector nimmt diese ungelösten Hashes jetzt in Software-Stücklistenexporte (SBOM) auf. Diese Pakete können zwar nicht auf Sicherheitslücken gescannt werden, ihre Hashwerte sind jedoch in der exportierten Komponentenliste verfügbar.

> **Tip**
>
> Seit dem 28. Februar 2026 unterstützt Amazon Inspector jetzt den Export von SBOMs für agentenlose Amazon EC2 EC2-Windows-Instances. Das Exportieren von SBOMs für agentenbasierte Amazon EC2 EC2-Windows-Instances wird derzeit nicht unterstützt.

Amazon Inspector Inspector-Formate

Amazon Inspector unterstützt den Export SBOMs in den mit CyclonedX 1.4 und SPDX 2.3 kompatiblen Formaten. Amazon Inspector exportiert SBOMs als `JSON` Dateien in den Amazon S3 S3-Bucket Ihrer Wahl.

> **Note**
>
> Exporte im SPDX-Format von Amazon Inspector sind mit Systemen kompatibel, die SPDX 2.3 verwenden, enthalten jedoch nicht das Creative Commons Zero (CC0) -Feld. Dies liegt

daran, dass die Aufnahme dieses Felds es Benutzern ermöglichen würde, das Material weiterzuverteilen oder zu bearbeiten.

Beispiel für das CyclonedX 1.4 SBOM-Format von Amazon Inspector


```
                    {
  "bomFormat": "CycloneDX",
  "specVersion": "1.4",
  "version": 1,
  "metadata": {
    "timestamp": "2023-06-02T01:17:46Z",
    "component": null,
    "properties": [
      {
        "name": "imageId",
        "value":
 "sha256:c8ee97f7052776ef223080741f61fcdf6a3a9107810ea9649f904aa4269fdac6"
      },
      {
        "name": "architecture",
        "value": "arm64"
      },
      {
        "name": "accountId",
        "value": "111122223333"
      },
      {
        "name": "resourceType",
        "value": "AWS_ECR_CONTAINER_IMAGE"
      }
    ]
  },
  "components": [
    {
      "type": "library",
      "name": "pip",
      "purl": "pkg:pypi/pip@22.0.4?path=usr/local/lib/python3.8/site-packages/
pip-22.0.4.dist-info/METADATA",
      "bom-ref": "98dc550d1e9a0b24161daaa0d535c699"
    },
    {
```

```
      "type": "application",
      "name": "libss2",
      "purl": "pkg:dpkg/libss2@1.44.5-1+deb10u3?
arch=ARM64&epoch=0&upstream=libss2-1.44.5-1+deb10u3.src.dpkg",
      "bom-ref": "2f4d199d4ef9e2ae639b4f8d04a813a2"
    },
    {
      "type": "application",
      "name": "liblz4-1",
      "purl": "pkg:dpkg/liblz4-1@1.8.3-1+deb10u1?
arch=ARM64&epoch=0&upstream=liblz4-1-1.8.3-1+deb10u1.src.dpkg",
      "bom-ref": "9a6be8907ead891b070e60f5a7b7aa9a"
    },
    {
      "type": "application",
      "name": "mawk",
      "purl": "pkg:dpkg/mawk@1.3.3-17+b3?
arch=ARM64&epoch=0&upstream=mawk-1.3.3-17+b3.src.dpkg",
      "bom-ref": "c2015852a729f97fde924e62a16f78a5"
    },
    {
      "type": "application",
      "name": "libgmp10",
      "purl": "pkg:dpkg/libgmp10@6.1.2+dfsg-4+deb10u1?
arch=ARM64&epoch=2&upstream=libgmp10-6.1.2+dfsg-4+deb10u1.src.dpkg",
      "bom-ref": "52907290f5beef00dff8da77901b1085"
    },
    {
      "type": "application",
      "name": "ncurses-bin",
      "purl": "pkg:dpkg/ncurses-bin@6.1+20181013-2+deb10u3?
arch=ARM64&epoch=0&upstream=ncurses-bin-6.1+20181013-2+deb10u3.src.dpkg",
      "bom-ref": "cd20cfb9ebeeadba3809764376f43bce"
    }
  ],
  "vulnerabilities": [
    {
      "id": "CVE-2022-40897",
      "affects": [
        {
          "ref": "a74a4862cc654a2520ec56da0c81cdb3"
        },
        {
          "ref": "0119eb286405d780dc437e7dbf2f9d9d"
```

```
        }
      ]
    }
  ]
}
```

Beispiel für das SPDX 2.3 SBOM-Format von Amazon Inspector

```
{
 "name": "409870544328/EC2/i-022fba820db137c64/ami-074ea14c08effb2d8",
 "spdxVersion": "SPDX-2.3",
 "creationInfo": {
  "created": "2023-06-02T21:19:22Z",
  "creators": [
   "Organization: 409870544328",
   "Tool: Amazon Inspector SBOM Generator"
  ]
 },
 "documentNamespace": "EC2://i-022fba820db137c64/AMAZON_LINUX_2/null/x86_64",
 "comment": "",
 "packages": [{
   "name": "elfutils-libelf",
   "versionInfo": "0.176-2.amzn2",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
    "referenceCategory": "PACKAGE-MANAGER",
    "referenceType": "purl",
    "referenceLocator": "pkg:rpm/elfutils-libelf@0.176-2.amzn2?
arch=X86_64&epoch=0&upstream=elfutils-libelf-0.176-2.amzn2.src.rpm"
   }],
   "SPDXID": "SPDXRef-Package-rpm-elfutils-libelf-ddf56a513c0e76ab2ae3246d9a91c463"
  },
  {
   "name": "libcurl",
   "versionInfo": "7.79.1-1.amzn2.0.1",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
```

```
     "referenceCategory": "PACKAGE-MANAGER",
     "referenceType": "purl",
     "referenceLocator": "pkg:rpm/libcurl@7.79.1-1.amzn2.0.1?
arch=X86_64&epoch=0&upstream=libcurl-7.79.1-1.amzn2.0.1.src.rpm"
    },
    {
     "referenceCategory": "SECURITY",
     "referenceType": "vulnerability",
     "referenceLocator": "CVE-2022-32205"
    }
   ],
   "SPDXID": "SPDXRef-Package-rpm-libcurl-710fb33829bc5106559bcd380cddb7d5"
  },
  {
   "name": "hunspell-en-US",
   "versionInfo": "0.20121024-6.amzn2.0.1",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
    "referenceCategory": "PACKAGE-MANAGER",
    "referenceType": "purl",
    "referenceLocator": "pkg:rpm/hunspell-en-US@0.20121024-6.amzn2.0.1?
arch=NOARCH&epoch=0&upstream=hunspell-en-US-0.20121024-6.amzn2.0.1.src.rpm"
   }],
   "SPDXID": "SPDXRef-Package-rpm-hunspell-en-US-de19ae0883973d6cea5e7e079d544fe5"
  },
  {
   "name": "grub2-tools-minimal",
   "versionInfo": "2.06-2.amzn2.0.6",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
     "referenceCategory": "PACKAGE-MANAGER",
     "referenceType": "purl",
     "referenceLocator": "pkg:rpm/grub2-tools-minimal@2.06-2.amzn2.0.6?
arch=X86_64&epoch=1&upstream=grub2-tools-minimal-2.06-2.amzn2.0.6.src.rpm"
    },
    {
     "referenceCategory": "SECURITY",
     "referenceType": "vulnerability",
     "referenceLocator": "CVE-2021-3981"
    }
```

```
   ],
   "SPDXID": "SPDXRef-Package-rpm-grub2-tools-minimal-c56b7ea76e5a28ab8f232ef6d7564636"
  },
  {
   "name": "unixODBC-devel",
   "versionInfo": "2.3.1-14.amzn2",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
    "referenceCategory": "PACKAGE-MANAGER",
    "referenceType": "purl",
    "referenceLocator": "pkg:rpm/unixODBC-devel@2.3.1-14.amzn2?
arch=X86_64&epoch=0&upstream=unixODBC-devel-2.3.1-14.amzn2.src.rpm"
   }],
   "SPDXID": "SPDXRef-Package-rpm-unixODBC-devel-1bb35add92978df021a13fc9f81237d2"
  }
 ],
 "relationships": [{
   "spdxElementId": "SPDXRef-DOCUMENT",
   "relatedSpdxElement": "SPDXRef-Package-rpm-elfutils-libelf-
ddf56a513c0e76ab2ae3246d9a91c463",
   "relationshipType": "DESCRIBES"
  },
  {
   "spdxElementId": "SPDXRef-DOCUMENT",
   "relatedSpdxElement": "SPDXRef-Package-rpm-yajl-8476ce2db98b28cfab2b4484f84f1903",
   "relationshipType": "DESCRIBES"
  },
  {
   "spdxElementId": "SPDXRef-DOCUMENT",
   "relatedSpdxElement": "SPDXRef-Package-rpm-unixODBC-
devel-1bb35add92978df021a13fc9f81237d2",
   "relationshipType": "DESCRIBES"
  }
 ],
 "SPDXID": "SPDXRef-DOCUMENT"
}
```

Filtert für SBOMs

Beim Exportieren können SBOMs Sie Filter einbeziehen, um Berichte für bestimmte Teilmengen von Ressourcen zu erstellen. Wenn Sie keinen Filter angeben, werden die SBOMs für alle aktiven, unterstützten Ressourcen exportiert. Und wenn Sie ein delegierter Administrator sind, umfasst dies auch Ressourcen für alle Mitglieder. Die folgenden Filter sind verfügbar:

- AccountID — Dieser Filter kann SBOMs für den Export aller Ressourcen verwendet werden, die mit einer bestimmten Konto-ID verknüpft sind.
- EC2-Instance-Tag — Dieser Filter kann verwendet werden, um SBOMs für EC2-Instances mit bestimmten Tags zu exportieren.
- Funktionsname — Dieser Filter kann SBOMs für den Export bestimmter Lambda-Funktionen verwendet werden.
- Bild-Tag — Dieser Filter kann SBOMs für den Export von Container-Images mit bestimmten Tags verwendet werden.
- Lambda-Funktions-Tag — Dieser Filter kann SBOMs für den Export von Lambda-Funktionen mit bestimmten Tags verwendet werden.
- Ressourcentyp — Dieser Filter kann verwendet werden, um den Ressourcentyp zu filtern: EC2/ECR/Lambda.
- Ressourcen-ID — Dieser Filter kann verwendet werden, um eine SBOM für eine bestimmte Ressource zu exportieren.
- Repository-Name — Dieser Filter kann verwendet werden, um Container-Images in bestimmten Repositorys zu generieren SBOMs .

Konfigurieren und exportieren SBOMs

Für den Export SBOMs müssen Sie zunächst einen Amazon S3 S3-Bucket und einen AWS KMS Schlüssel konfigurieren, den Amazon Inspector verwenden darf. Sie können Filter verwenden, um SBOMs für bestimmte Teilmengen Ihrer Ressourcen zu exportieren. Um SBOMs für mehrere Konten in einer AWS Organisation zu exportieren, folgen Sie diesen Schritten, während Sie als delegierter Amazon Inspector-Administrator angemeldet sind.

Voraussetzungen

- Unterstützte Ressourcen, die aktiv von Amazon Inspector überwacht werden.

- Ein Amazon S3 S3-Bucket, der mit einer Richtlinie konfiguriert ist, die es Amazon Inspector ermöglicht, Objekte hinzuzufügen. Informationen zur Konfiguration der Richtlinie finden Sie unter Exportberechtigungen konfigurieren.
- Ein AWS KMS Schlüssel, der mit einer Richtlinie konfiguriert ist, die es Amazon Inspector ermöglicht, Ihre Berichte zu verschlüsseln. Informationen zur Konfiguration der Richtlinie finden Sie unter Einen AWS KMS Schlüssel für den Export konfigurieren.

> **Note**
>
> Wenn Sie zuvor einen Amazon S3 S3-Bucket und einen AWS KMS Schlüssel für den Ergebnisexport konfiguriert haben, können Sie denselben Bucket und Schlüssel für den SBOM-Export verwenden.

Wählen Sie Ihre bevorzugte Zugriffsmethode für den Export einer SBOM.

Console

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mit der AWS-Region Auswahl in der oberen rechten Ecke der Seite die Region mit den Ressourcen aus, für die Sie SBOM exportieren möchten.
3. Wählen Sie im Navigationsbereich die Option Export (Exportieren) SBOMs aus.
4. (Optional) Verwenden Sie auf der SBOMsExportseite das Menü Filter hinzufügen, um eine Teilmenge von Ressourcen auszuwählen, für die Berichte erstellt werden sollen. Wenn kein Filter angegeben ist, exportiert Amazon Inspector Berichte für alle aktiven Ressourcen. Wenn Sie ein delegierter Administrator sind, umfasst dies alle aktiven Ressourcen in Ihrer Organisation.
5. Wählen Sie unter Exporteinstellung das gewünschte Format für die SBOM aus.
6. Geben Sie eine Amazon S3-URI ein oder wählen Sie Amazon S3 durchsuchen, um einen Amazon S3 S3-Standort zum Speichern der SBOM auszuwählen.
7. Geben Sie einen AWS KMS Schlüssel ein, der für Amazon Inspector konfiguriert ist, um Ihre Berichte zu verschlüsseln.

API

- Verwenden Sie den CreateSbomExportBetrieb der Amazon Inspector Inspector-API, um Ihre Ressourcen programmgesteuert zu exportieren SBOMs .

  Verwenden Sie in Ihrer Anfrage den `reportFormat` Parameter, um das SBOM-Ausgabeformat anzugeben, und wählen Sie oder. `CYCLONEDX_1_4` `SPDX_2_3` Der `s3Destination` Parameter ist erforderlich, und Sie müssen einen S3-Bucket angeben, der mit einer Richtlinie konfiguriert ist, die es Amazon Inspector ermöglicht, in diesen Bucket zu schreiben. Verwenden Sie optional `resourceFilterCriteria` Parameter, um den Umfang des Berichts auf bestimmte Ressourcen zu beschränken.

AWS CLI

- AWS Command Line Interface Führen Sie den folgenden Befehl aus, um SBOMs für Ihre Ressourcen mit dem folgenden Befehl zu exportieren:

  ```
  aws inspector2 create-sbom-export --report-format
  FORMAT --s3-destination bucketName=amzn-s3-demo-
  bucket1,keyPrefix=PREFIX,kmsKeyArn=arn:aws:kms:Region:111122223333:key/123
  ```

  Ersetzen Sie es in Ihrer Anfrage *FORMAT* durch das Format Ihrer Wahl, `CYCLONEDX_1_4` oder`SPDX_2_3`. Ersetzen Sie dann das *user input placeholders* für das S3-Ziel durch den Namen des S3-Buckets, in den exportiert werden soll, das Präfix, das für die Ausgabe in S3 verwendet werden soll, und den ARN für den KMS-Schlüssel, den Sie zum Verschlüsseln der Berichte verwenden.

EventBridge Amazon-Ereignisschema für Amazon Inspector-Ereignisse

Amazon EventBridge liefert einen Stream von Echtzeitdaten aus Anwendungen und anderen Anwendungen AWS-Services an Ziele wie AWS Lambda Funktionen, Amazon Simple Notification Service-Themen und Datenstreams in Amazon Kinesis Data Streams. Um die Integration mit anderen Anwendungen, Diensten und Systemen zu unterstützen, veröffentlicht Amazon Inspector die Ergebnisse automatisch EventBridge als Ereignisse. Sie können Amazon Inspector verwenden, um Ereignisse für Ergebnisse, Berichterstattung und Scans zu veröffentlichen. Dieser Abschnitt enthält Beispielschemas für EventBridge Ereignisse.

Themen

- EventBridge Amazon-Basisschema für Amazon Inspector
- Beispiel für das Auffinden von Ereignissen in Amazon Inspector
- Beispiel für ein vollständiges Amazon Inspector Inspector-Ereignisschema für den ersten Scan
- Beispiel für ein Amazon Inspector Inspector-Abdeckungsereignisschema
- Beispiel für ein Schema zur auto Aktivierung von Amazon Inspector

EventBridge Amazon-Basisschema für Amazon Inspector

Das Folgende ist ein Beispiel für das grundlegende Schema für ein EventBridge Ereignis für Amazon Inspector. Die Veranstaltungsdetails unterscheiden sich je nach Art des Ereignisses.

```
{
    "version": "0",
    "id": "Event ID",
    "detail-type": "Inspector2 *event type*",
    "source": "aws.inspector2",
    "account": "AWS-Konto ID (string)",
    "time": "event timestamp (string)",
    "region": "AWS-Region (string)",
    "resources": [
        *IDs or ARNs of the resources involved in the event*
    ],
    "detail": {
        *Details of an Amazon Inspector event type*
```

```
    }
}
```

Beispiel für das Auffinden von Ereignissen in Amazon Inspector

Im Folgenden finden Sie Beispiele für das Schema für ein EventBridge Ereignis mit Ergebnissen von Amazon Inspector. Findungsereignisse werden ausgelöst, wenn Amazon Inspector eine Softwareschwachstelle oder ein Netzwerkproblem in einer Ihrer Ressourcen identifiziert. Eine Anleitung zum Erstellen von Benachrichtigungen als Reaktion auf diese Art von Ereignis finden Sie unterErstellung benutzerdefinierter Antworten auf Amazon Inspector-Ergebnisse mit Amazon EventBridge.

Die folgenden Felder identifizieren ein Findereignis:

- ist `detail-type` auf `Inspector2 Finding` gesetzt.
- `detail`beschreibt den Befund.
- `detail.resources.tags`ist der Ort, an dem Schlüsselwertdaten gespeichert werden.

Jedes Objekt `detail.resources` kann auch die folgenden Felder enthalten. Diese Felder sind für alle Ressourcen vorhanden, auch für AWS Ressourcen.

Bei AWS Ressourcen werden diese Felder neben den AWS-spezifischen Feldern angezeigt. Sie können sich auf eine einzige, anbieterunabhängige Darstellung verlassen, unabhängig davon, wo die Ressource gehostet wird.

- `provider`ist der Cloud-Anbieter der Ressource, zum Beispiel oder. `AWS` `AZURE`
- `providerAccountId`dist die Konto-ID des Cloud-Anbieters für die Ressource, z. B. eine Azure-Abonnement-ID. Bei AWS Ressourcen entspricht dies dem `account` Feld.
- `providerOrgId`dist die Organisations- oder Mandanten-ID des Cloud-Anbieters für die Ressource, z. B. eine Azure-Mandanten-ID. Amazon Inspector füllt dieses Feld für `AZURE` Ressourcen aus.
- `detail.resources.details`enthält ein anbieterunabhängiges Ressourcendetailobjekt, das der Ressource entsprichttype: `vm` für eine virtuelle Maschine, für ein Container-Image oder `image` `serverlessFunction` für eine serverlose Funktion. Dies sind die Multi-Cloud-Äquivalente der Objekte, und. `awsEc2Instance` `awsEcrContainerImage` `awsLambdaFunction` Für AWS Ressourcen schließt Amazon Inspector neben dem entsprechenden -spezifischen Objekt auch das anbieterunabhängige Objekt ein. AWS Ein Amazon EC2 EC2-Instance-Ergebnis umfasst beispielsweise `awsEc2Instance` sowohl `vm` als auch.

Sie können die Tabs filtern, um nach Ereignisschemas für verschiedene Ressourcen und Suchtypen zu suchen.

Amazon EC2 package vulnerability finding


```
{
    "version": "0",
    "id": "4d621919-f1f4-4201-a0e2-37e4e330ff51",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T17:00:36Z",
    "region": "eu-central-1",
    "resources": [
        "i-12345678901234567"
    ],
    "detail": {
        "awsAccountId": "123456789012",
        "description": "In snapd versions prior to 2.62, snapd failed to properly
 check the destination of symbolic links when extracting a snap. The snap format
 is a squashfs file-system image and so can contain symbolic links and other file
 types. Various file entries within the snap squashfs image (such as icons and
 desktop files etc) are directly read by snapd when it is extracted. An attacker who
 could convince a user to install a malicious snap which contained symbolic links
 at these paths could then cause snapd to write out the contents of the symbolic
 link destination into a world-readable directory. This in-turn could allow an
 unprivileged user to gain access to privileged information.",
        "epss": {
            "score": 0.00043
        },
        "exploitAvailable": "NO",
        "findingArn": "arn:aws:inspector2:eu-
central-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:59:44.356 UTC 2024",
        "fixAvailable": "YES",
        "inspectorScore": 4.8,
        "inspectorScoreDetails": {
            "adjustedCvss": {
                "adjustments": [],
                "cvssSource": "UBUNTU_CVE",
                "score": 4.8,
                "scoreSource": "UBUNTU_CVE",
```

```
                "scoringVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:L",
                "version": "3.1"
            }
        },
        "lastObservedAt": "Wed Sep 04 16:59:44.476 UTC 2024",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 4.8,
                    "scoringVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:L",
                    "source": "UBUNTU_CVE",
                    "version": "3.1"
                },
                {
                    "baseScore": 7.3,
                    "scoringVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H",
                    "source": "NVD",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [
                "https://www.cve.org/CVERecord?id=CVE-2024-29069",
                "https://ubuntu.com/security/notices/USN-6940-1"
            ],
            "relatedVulnerabilities": [
                "USN-6940-1"
            ],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2024/
CVE-2024-29069.html",
            "vendorCreatedAt": "Thu Jul 25 20:15:00.000 UTC 2024",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2024-29069",
            "vulnerablePackages": [
                {
                    "arch": "ALL",
                    "epoch": 0,
                    "fixedInVersion": "0:2.63+22.04ubuntu0.1",
                    "name": "snapd",
                    "packageManager": "OS",
                    "remediation": "apt-get update && apt-get upgrade",
                    "version": "2.63"
                }
            ]
```

```
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
                "details": {
                    "awsEc2Instance": {
                        "iamInstanceProfileArn":
 "arn:aws:iam::123456789012:instance-profile/AmazonSSMRoleForInstancesQuickSetup",
                        "imageId": "ami-02ff980600c693b38",
                        "ipV4Addresses": [
                            "192.0.2.10",
                            "203.0.113.20"
                        ],
                        "ipV6Addresses": [],
                        "launchedAt": "Wed Sep 04 16:57:40.000 UTC 2024",
                        "platform": "UBUNTU_22_04",
                        "subnetId": "subnet-12345678",
                        "type": "t2.small",
                        "vpcId": "vpc-12345678"
                    },
                    "vm": {
                        "type": "t2.small",
                        "vmImageReference": "ami-02ff980600c693b38",
                        "ipV4Addresses": [
                            "192.0.2.10",
                            "203.0.113.20"
                        ],
                        "ipV6Addresses": [],
                        "networkId": "vpc-12345678",
                        "subnetIds": [
                            "subnet-12345678"
                        ],
                        "launchedAt": "Wed Sep 04 16:57:40.000 UTC 2024",
                        "platform": "UBUNTU_22_04",
                        "executionRole": "arn:aws:iam::123456789012:instance-
profile/AmazonSSMRoleForInstancesQuickSetup"
                    }
                },
                "id": "i-12345678901234567",
                "partition": "aws",
```

```
                "provider": "AWS",
                "providerAccountId": "123456789012",
                "region": "eu-central-1",
                "type": "AWS_EC2_INSTANCE"
            }
        ],
        "severity": "MEDIUM",
        "status": "CLOSED",
        "title": "CVE-2024-29069 - snapd",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 17:00:36.951 UTC 2024"
    }
}
```

Amazon EC2 network reachability finding

```
{
    "version": "0",
    "id": "9eb1603b-4263-19ec-8be2-33184694cb92",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-05T13:06:56Z",
    "region": "eu-central-1",
    "resources": ["i-12345678901234567"],
    "detail": {
        "awsAccountId": "123456789012",
        "description": "On the instance i-12345678901234567, the port range
 22-22 is reachable from the InternetGateway igw-261bab4d from an attached ENI
 eni-094ad651219472857.",
        "findingArn": "arn:aws:inspector2:eu-
central-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Thu Sep 05 13:06:56.334 UTC 2024",
        "lastObservedAt": "Thu Sep 05 13:06:56.334 UTC 2024",
        "networkReachabilityDetails": {
            "networkPath": {
                "steps": [{
                    "componentId": "igw-261bab4d",
                    "componentType": "AWS::EC2::InternetGateway"
                }, {
                    "componentId": "acl-171b527d",
```

```
                    "componentType": "AWS::EC2::NetworkAcl"
                }, {
                    "componentId": "sg-0d34debf87410f2d9",
                    "componentType": "AWS::EC2::SecurityGroup"
                }, {
                    "componentId": "eni-094ad651219472857",
                    "componentType": "AWS::EC2::NetworkInterface"
                }, {
                    "componentId": "i-12345678901234567",
                    "componentType": "AWS::EC2::Instance"
                }]
            },
            "openPortRange": {
                "begin": 22,
                "end": 22
            },
            "protocol": "TCP"
        },
        "remediation": {
            "recommendation": {
                "text": "You can restrict access to your instance by modifying the
 Security Groups or ACLs in the network path."
            }
        },
        "resources": [{
            "details": {
                "awsEc2Instance": {
                    "iamInstanceProfileArn": "arn:aws:iam::123456789012:instance-
profile/AmazonSSMRoleForInstancesQuickSetup",
                    "imageId": "ami-02ff980600c693b38",
                    "ipV4Addresses": ["192.0.2.10", "203.0.113.20"],
                    "ipV6Addresses": [],
                    "launchedAt": "Wed Sep 04 17:41:24.000 UTC 2024",
                    "platform": "UBUNTU_22_04",
                    "subnetId": "subnet-12345678",
                    "type": "t2.small",
                    "vpcId": "vpc-12345678"
                },
                "vm": {
                    "type": "t2.small",
                    "vmImageReference": "ami-02ff980600c693b38",
                    "ipV4Addresses": ["192.0.2.10", "203.0.113.20"],
                    "ipV6Addresses": [],
                    "networkId": "vpc-12345678",
```

```
                    "subnetIds": ["subnet-12345678"],
                    "launchedAt": "Wed Sep 04 17:41:24.000 UTC 2024",
                    "platform": "UBUNTU_22_04",
                    "executionRole": "arn:aws:iam::123456789012:instance-profile/
AmazonSSMRoleForInstancesQuickSetup"
                }
            },
            "id": "i-12345678901234567",
            "partition": "aws",
            "provider": "AWS",
            "providerAccountId": "123456789012",
            "region": "eu-central-1",
            "type": "AWS_EC2_INSTANCE"
        }],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "Port 22 is reachable from an Internet Gateway - TCP",
        "type": "NETWORK_REACHABILITY",
        "updatedAt": "Thu Sep 05 13:06:56.334 UTC 2024"
    }
}
```

Amazon ECR package vulnerability finding

```
{
    "version": "0",
    "id": "5325facf-a1aa-7d97-6bce-25fde6f6d2fc",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T16:55:38Z",
    "region": "eu-central-1",
    "resources": [
        "arn:aws:ecr:eu-central-1:123456789012:repository/inspector2/
sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d"
    ],
    "detail.resources.tags.testkey": "allow",
    "detail": {
        "awsAccountId": "123456789012",
        "description": "Possible denial of service in X.509 name checks",
        "epss": {
```

```
            "score": 0.00045
        },
        "exploitAvailable": "NO",
        "findingArn": "arn:aws:inspector2:eu-
central-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:55:38.411 UTC 2024",
        "fixAvailable": "YES",
        "lastObservedAt": "Wed Sep 04 16:55:38.411 UTC 2024",
        "packageVulnerabilityDetails": {
            "cvss": [],
            "referenceUrls": [
                "https://www.cve.org/CVERecord?id=CVE-2024-6119",
                "https://ubuntu.com/security/notices/USN-6986-1"
            ],
            "relatedVulnerabilities": [
                "USN-6986-1"
            ],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2024/
CVE-2024-6119.html",
            "vendorCreatedAt": "Tue Sep 03 00:00:00.000 UTC 2024",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2024-6119",
            "vulnerablePackages": [
                {
                    "arch": "ARM64",
                    "epoch": 0,
                    "fixedInVersion": "0:3.0.13-0ubuntu3.4",
                    "name": "libssl3t64",
                    "packageManager": "OS",
                    "release": "0ubuntu3.2",
                    "remediation": "apt-get update && apt-get upgrade",
                    "sourceLayerHash":
 "sha256:1567e7ea90b67fc95ccdeeec39bdc3045098dee7e0c604975b957a9f8c0e9616",
                    "version": "3.0.13"
                },
                {
                    "arch": "ARM64",
                    "epoch": 0,
                    "fixedInVersion": "0:3.0.13-0ubuntu3.4",
                    "name": "openssl",
                    "packageManager": "OS",
                    "release": "0ubuntu3.2",
                    "remediation": "apt-get update && apt-get upgrade",
```

```
                    "sourceLayerHash":
 "sha256:1567e7ea90b67fc95ccdeeec39bdc3045098dee7e0c604975b957a9f8c0e9616",
                    "version": "3.0.13"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
                "details": {
                    "awsEcrContainerImage": {
                        "architecture": "arm64",
                        "imageHash":
 "sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d",
                        "imageTags": [
                            "ubuntu_latest"
                        ],
                        "platform": "UBUNTU_24_04",
                        "pushedAt": "Wed Sep 04 16:55:28.000 UTC 2024",
                        "registry": "123456789012",
                        "repositoryName": "inspector2"
                    },
                    "image": {
                        "repositoryName": "inspector2",
                        "registry": "123456789012",
                        "imageTags": [
                            "ubuntu_latest"
                        ],
                        "imageDigest":
 "sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d",
                        "pushedAt": "Wed Sep 04 16:55:28.000 UTC 2024",
                        "architecture": "arm64",
                        "platform": "UBUNTU_24_04"
                    }
                },
                "id": "arn:aws:ecr:eu-central-1:123456789012:repository/inspector2/
sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d",
                "partition": "aws",
                "provider": "AWS",
                "providerAccountId": "123456789012",
```

```
                "region": "eu-central-1",
                "type": "AWS_ECR_CONTAINER_IMAGE"
            }
        ],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "CVE-2024-6119 - libssl3t64, openssl",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 16:55:38.411 UTC 2024"
    }
}
```


Lambda package vulnerability finding


```
{
    "version": "0",
    "id": "9eadd71a-e49c-9864-6ba9-2a5d3f83c88f",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T16:50:37Z",
    "region": "eu-central-1",
    "resources": [
        "arn:aws:lambda:eu-central-1:123456789012:function:VulnerableFunction:
$LATEST"
    ],
    "detail": {
        "awsAccountId": "123456789012",
        "description": "Flask is a lightweight WSGI web application framework. When
 all of the following conditions are met, a response containing data intended for
 one client may be cached and subsequently sent by the proxy to other clients. If
 the proxy also caches `Set-Cookie` headers, it may send one client's `session`
 cookie to other clients. The severity depends on the application's use of the
 session and the proxy's behavior regarding cookies. The risk depends on all these
 conditions being met.\n\n1. The application must be hosted behind a caching proxy
 that does not strip cookies or ignore responses with cookies. 2. The application
 sets `session.permanent = True` 3. The application does not access or modify the
 session at any point during a request. 4. `SESSION_REFRESH_EACH_REQUEST` enabled
 (the default). 5. The application does not set a `Cache-Control` header to indicate
 that a page is private or should not be cached.\n\nThis happens because vulnerable
 versions of Flask only set the `Vary: Cookie` header when the session is ac",
```

```
        "epss": {
            "score": 0.00208
        },
        "exploitAvailable": "YES",
        "exploitabilityDetails": {
            "lastKnownExploitAt": "Sat Aug 31 00:04:50.000 UTC 2024"
        },
        "findingArn": "arn:aws:inspector2:eu-
central-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:50:37.627 UTC 2024",
        "fixAvailable": "YES",
        "inspectorScore": 7.5,
        "inspectorScoreDetails": {
            "adjustedCvss": {
                "cvssSource": "NVD",
                "score": 7.5,
                "scoreSource": "NVD",
                "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N",
                "version": "3.1"
            }
        },
        "lastObservedAt": "Wed Sep 04 16:50:37.627 UTC 2024",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 7.5,
                    "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N",
                    "source": "NVD",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [
                "https://www.debian.org/security/2023/dsa-5442",
                "https://lists.debian.org/debian-lts-announce/2023/08/msg00024.html"
            ],
            "relatedVulnerabilities": [],
            "source": "NVD",
            "sourceUrl": "https://nvd.nist.gov/vuln/detail/CVE-2023-30861",
            "vendorCreatedAt": "Tue May 02 18:15:52.000 UTC 2023",
            "vendorSeverity": "HIGH",
            "vendorUpdatedAt": "Sun Aug 20 21:15:09.000 UTC 2023",
            "vulnerabilityId": "CVE-2023-30861",
            "vulnerablePackages": [
                {
```

```
                    "epoch": 0,
                    "filePath": "requirements.txt",
                    "fixedInVersion": "2.3.2",
                    "name": "flask",
                    "packageManager": "PIP",
                    "version": "2.0.0"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
                "details": {
                    "awsLambdaFunction": {
                        "architectures": [
                            "X86_64"
                        ],
                        "codeSha256": "O7jkFEmfPB+CK3Y6Pby5zW9gjG
+zusAaqRRMGS8B27c=",
                        "executionRoleArn": "arn:aws:iam::123456789012:role/service-
role/VulnerableFunction-role-f9vs5mq8",
                        "functionName": "VulnerableFunction",
                        "lastModifiedAt": "Wed Sep 04 16:50:20.000 UTC 2024",
                        "packageType": "ZIP",
                        "runtime": "PYTHON_3_11",
                        "version": "$LATEST"
                    },
                    "serverlessFunction": {
                        "serverlessFunctionName": "VulnerableFunction",
                        "runtime": "PYTHON_3_11",
                        "version": "$LATEST",
                        "codeDigest": "O7jkFEmfPB+CK3Y6Pby5zW9gjG
+zusAaqRRMGS8B27c=",
                        "lastModifiedAt": "Wed Sep 04 16:50:20.000 UTC 2024",
                        "executionRole": "arn:aws:iam::123456789012:role/service-
role/VulnerableFunction-role-f9vs5mq8",
                        "packageType": "ZIP",
                        "architectures": [
                            "X86_64"
                        ]
```

```
                    }
                },
                "id": "arn:aws:lambda:eu-
central-1:123456789012:function:VulnerableFunction:$LATEST",
                "partition": "aws",
                "provider": "AWS",
                "providerAccountId": "123456789012",
                "region": "eu-central-1",
                "type": "AWS_LAMBDA_FUNCTION"
            }
        ],
        "severity": "HIGH",
        "status": "ACTIVE",
        "title": "CVE-2023-30861 - flask",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 16:50:37.627 UTC 2024"
    }
}
```

Lambda code vulnerability finding

```
{
    "version": "0",
    "id": "e764f7be-f931-ff1b-204b-8cab2d91724b",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T16:51:01Z",
    "region": "eu-central-1",
    "resources": [
        "arn:aws:lambda:eu-central-1:123456789012:function:VulnerableFunction:
$LATEST"
    ],
    "detail": {
        "awsAccountId": "123456789012",
        "codeVulnerabilityDetails": {
            "cwes": [
                "CWE-798"
            ],
            "detectorId": "python/hardcoded-credentials@v1.0",
            "detectorName": "Hardcoded credentials",
```

```
            "detectorTags": [
                "secrets",
                "security",
                "owasp-top10",
                "top25-cwes",
                "cwe-798",
                "Python"
            ],
            "filePath": {
                "endLine": 6,
                "fileName": "lambda_function.py",
                "filePath": "lambda_function.py",
                "startLine": 6
            },
            "ruleId": "python-detect-hardcoded-aws-credentials"
        },
        "description": "Access credentials, such as passwords and access keys,
 should not be hardcoded in source code. Hardcoding credentials may cause leaks even
 after removing them. This is because version control systems might retain older
 versions of the code. Credentials should be stored securely and obtained from the
 runtime environment.",
        "findingArn": "arn:aws:inspector2:eu-
central-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:51:01.869 UTC 2024",
        "lastObservedAt": "Wed Sep 04 16:51:01.869 UTC 2024",
        "remediation": {
            "recommendation": {
                "text": "Your code uses hardcoded AWS credentials which might
 allow unauthorized users access to your AWS account. These attacks can occur
 a long time after the credentials are removed from the code. We recommend that
 you set AWS credentials with environment variables or an AWS profile instead.
 You should consider deleting the affected account or rotating the secret key
 and then monitoring Amazon CloudWatch for unexpected activity.\n[https://
boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html](https://
boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html)"
            }
        },
        "resources": [
            {
                "details": {
                    "awsLambdaFunction": {
                        "architectures": [
                            "X86_64"
                        ],
```

```
                        "codeSha256": "O7jkFEmfPB+CK3Y6Pby5zW9gjG
+zusAaqRRMGS8B27c=",
                        "executionRoleArn": "arn:aws:iam::123456789012:role/service-
role/VulnerableFunction-role-f9vs5mq8",
                        "functionName": "VulnerableFunction",
                        "lastModifiedAt": "Wed Sep 04 16:50:20.000 UTC 2024",
                        "packageType": "ZIP",
                        "runtime": "PYTHON_3_11",
                        "version": "$LATEST"
                    },
                    "serverlessFunction": {
                        "serverlessFunctionName": "VulnerableFunction",
                        "runtime": "PYTHON_3_11",
                        "version": "$LATEST",
                        "codeDigest": "O7jkFEmfPB+CK3Y6Pby5zW9gjG
+zusAaqRRMGS8B27c=",
                        "lastModifiedAt": "Wed Sep 04 16:50:20.000 UTC 2024",
                        "executionRole": "arn:aws:iam::123456789012:role/service-
role/VulnerableFunction-role-f9vs5mq8",
                        "packageType": "ZIP",
                        "architectures": [
                            "X86_64"
                        ]
                    }
                },
                "id": "arn:aws:lambda:eu-
central-1:123456789012:function:VulnerableFunction:$LATEST",
                "partition": "aws",
                "provider": "AWS",
                "providerAccountId": "123456789012",
                "region": "eu-central-1",
                "type": "AWS_LAMBDA_FUNCTION"
            }
        ],
        "severity": "CRITICAL",
        "status": "ACTIVE",
        "title": "CWE-798 - Hardcoded credentials",
        "type": "CODE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 16:51:01.869 UTC 2024"
    }
}
```

Azure virtual machine package vulnerability finding


```
{
    "version": "0",
    "id": "a1b2c3d4-1111-2222-3333-444455556666",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T17:00:36Z",
    "region": "us-east-1",
    "resources": [
        "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/resourcegroups/example-
rg/providers/microsoft.compute/virtualmachines/example-vm"
    ],
    "detail": {
        "awsAccountId": "123456789012",
        "description": "A package vulnerability was detected on an Azure virtual
 machine monitored by Amazon Inspector.",
        "exploitAvailable": "NO",
        "findingArn": "arn:aws:inspector2:us-
east-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:59:44.356 UTC 2024",
        "fixAvailable": "YES",
        "lastObservedAt": "Wed Sep 04 16:59:44.476 UTC 2024",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 4.8,
                    "scoringVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:L",
                    "source": "UBUNTU_CVE",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [
                "https://www.cve.org/CVERecord?id=CVE-2024-29069"
            ],
            "relatedVulnerabilities": [],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2024/
CVE-2024-29069.html",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2024-29069",
            "vulnerablePackages": [
```

```
                {
                    "arch": "ALL",
                    "epoch": 0,
                    "fixedInVersion": "0:2.63+22.04ubuntu0.1",
                    "name": "snapd",
                    "packageManager": "OS",
                    "remediation": "apt-get update && apt-get upgrade",
                    "version": "2.63"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
                "details": {
                    "vm": {
                        "type": "Standard_D2s_v3",
                        "vmName": "example-vm",
                        "vmImageReference": "Canonical:0001-com-ubuntu-server-
jammy:22_04-lts:latest",
                        "ipV4Addresses": [
                            "10.0.0.4"
                        ],
                        "ipV6Addresses": [],
                        "networkId": "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-
eeeeeeeeeeee/resourcegroups/example-rg/providers/microsoft.network/virtualnetworks/
example-vnet",
                        "subnetIds": [
                            "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/
resourcegroups/example-rg/providers/microsoft.network/virtualnetworks/example-vnet/
subnets/default"
                        ],
                        "securityGroupIds": [
                            "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/
resourcegroups/example-rg/providers/microsoft.network/networksecuritygroups/example-
nsg"
                        ],
                        "launchedAt": "Wed Sep 04 16:57:40.000 UTC 2024",
                        "platform": "UBUNTU_22_04",
```

```
                        "executionRole": "/subscriptions/aaaaaaaa-bbbb-cccc-
dddd-eeeeeeeeeeee/resourcegroups/example-rg/providers/microsoft.managedidentity/
userassignedidentities/example-identity",
                        "keyName": "example-key"
                    }
                },
                "id": "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/
resourcegroups/example-rg/providers/microsoft.compute/virtualmachines/example-vm",
                "provider": "AZURE",
                "providerAccountId": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
                "providerOrgId": "11111111-2222-3333-4444-555555555555",
                "region": "eastus",
                "type": "Microsoft.Compute/virtualMachines"
            }
        ],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "CVE-2024-29069 - snapd",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 17:00:36.951 UTC 2024"
    }
}
```

Azure container image package vulnerability finding

```
{
    "version": "0",
    "id": "b2c3d4e5-2222-3333-4444-555566667777",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T16:55:38Z",
    "region": "us-east-1",
    "resources": [
        "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/
resourcegroups/example-rg/providers/microsoft.containerregistry/
registries/exampleregistry/repositories/inspector2/images/
sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d"
    ],
    "detail": {
        "awsAccountId": "123456789012",
```

```
        "description": "A package vulnerability was detected in an Azure Container
 Registry image monitored by Amazon Inspector.",
        "exploitAvailable": "NO",
        "findingArn": "arn:aws:inspector2:us-
east-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:55:38.411 UTC 2024",
        "fixAvailable": "YES",
        "lastObservedAt": "Wed Sep 04 16:55:38.411 UTC 2024",
        "packageVulnerabilityDetails": {
            "cvss": [],
            "referenceUrls": [
                "https://www.cve.org/CVERecord?id=CVE-2024-6119"
            ],
            "relatedVulnerabilities": [],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2024/
CVE-2024-6119.html",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2024-6119",
            "vulnerablePackages": [
                {
                    "arch": "ARM64",
                    "epoch": 0,
                    "fixedInVersion": "0:3.0.13-0ubuntu3.4",
                    "name": "openssl",
                    "packageManager": "OS",
                    "remediation": "apt-get update && apt-get upgrade",
                    "version": "3.0.13"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
                "details": {
                    "image": {
                        "repositoryName": "inspector2",
                        "registry": "exampleregistry.azurecr.io",
                        "imageTags": [
                            "ubuntu_latest"
```

```
                        ],
                        "imageDigest":
 "sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d",
                        "pushedAt": "Wed Sep 04 16:55:28.000 UTC 2024",
                        "architecture": "arm64",
                        "author": "example-author",
                        "platform": "UBUNTU_24_04"
                    }
                },
                "id": "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/
resourcegroups/example-rg/providers/microsoft.containerregistry/
registries/exampleregistry/repositories/inspector2/images/
sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d",
                "provider": "AZURE",
                "providerAccountId": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
                "providerOrgId": "11111111-2222-3333-4444-555555555555",
                "region": "eastus",
                "type": "Microsoft.ContainerRegistry/registry/containerImage"
            }
        ],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "CVE-2024-6119 - openssl",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 16:55:38.411 UTC 2024"
    }
}
```

Azure function app package vulnerability finding

```
{
    "version": "0",
    "id": "c3d4e5f6-3333-4444-5555-666677778888",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-09-04T16:50:37Z",
    "region": "us-east-1",
    "resources": [
        "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/resourcegroups/example-
rg/providers/microsoft.web/sites/example-functionapp"
```

```
    ],
    "detail": {
        "awsAccountId": "123456789012",
        "description": "A package vulnerability was detected in an Azure function
 app monitored by Amazon Inspector.",
        "exploitAvailable": "YES",
        "findingArn": "arn:aws:inspector2:us-
east-1:123456789012:finding/FINDING_ID",
        "firstObservedAt": "Wed Sep 04 16:50:37.627 UTC 2024",
        "fixAvailable": "YES",
        "lastObservedAt": "Wed Sep 04 16:50:37.627 UTC 2024",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 7.5,
                    "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N",
                    "source": "NVD",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [],
            "relatedVulnerabilities": [],
            "source": "NVD",
            "sourceUrl": "https://nvd.nist.gov/vuln/detail/CVE-2023-30861",
            "vendorSeverity": "HIGH",
            "vulnerabilityId": "CVE-2023-30861",
            "vulnerablePackages": [
                {
                    "epoch": 0,
                    "filePath": "requirements.txt",
                    "fixedInVersion": "2.3.2",
                    "name": "flask",
                    "packageManager": "PIP",
                    "version": "2.0.0"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
```

```
                "details": {
                    "serverlessFunction": {
                        "serverlessFunctionName": "example-functionapp",
                        "runtime": "PYTHON_3_11",
                        "version": "1",
                        "codeDigest": "O7jkFEmfPB+CK3Y6Pby5zW9gjG
+zusAaqRRMGS8B27c=",
                        "lastModifiedAt": "Wed Sep 04 16:50:20.000 UTC 2024",
                        "executionRole": "/subscriptions/aaaaaaaa-bbbb-cccc-
dddd-eeeeeeeeeeee/resourcegroups/example-rg/providers/microsoft.managedidentity/
userassignedidentities/example-identity",
                        "packageType": "ZIP",
                        "architectures": [
                            "X86_64"
                        ]
                    }
                },
                "id": "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/
resourcegroups/example-rg/providers/microsoft.web/sites/example-functionapp",
                "provider": "AZURE",
                "providerAccountId": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
                "providerOrgId": "11111111-2222-3333-4444-555555555555",
                "region": "eastus",
                "type": "Microsoft.Web/sites"
            }
        ],
        "severity": "HIGH",
        "status": "ACTIVE",
        "title": "CVE-2023-30861 - flask",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Wed Sep 04 16:50:37.627 UTC 2024"
    }
}
```


> **Note**
>
> Der Detailwert gibt die JSON-Details eines einzelnen Ergebnisses als Objekt zurück. Es wird nicht die gesamte Syntax der Ergebnisantwort zurückgegeben, die mehrere Ergebnisse innerhalb eines Arrays unterstützt.

Beispiel für ein vollständiges Amazon Inspector Inspector-Ereignisschema für den ersten Scan

Im Folgenden finden Sie ein Beispiel für das EventBridge Ereignisschema für ein Amazon Inspector Inspector-Ereignis zum Abschluss eines ersten Scans. Dieses Ereignis wird ausgelöst, wenn Amazon Inspector einen ersten Scan einer Ihrer Ressourcen abschließt.

Die folgenden Felder kennzeichnen ein Ereignis, bei dem der erste Scan abgeschlossen wurde:

- Das `detail-type` Feld ist auf eingestelltInspector2 Scan.
- Das `detail` Objekt enthält ein `finding-severity-counts` Objekt, das die Anzahl der Ergebnisse in den entsprechenden Schweregradkategorien, wie, und,CRITICAL,,HIGH, detailliert beschreibtMEDIUM.

Wählen Sie eine der Optionen aus, um je nach Ressourcentyp unterschiedliche Ereignisschemas für den ersten Scan anzuzeigen.

Amazon EC2 instance initial scan


```
{
    "version": "0",
    "id": "28a46762-6ac8-6cc4-4f55-bc9ab99af928",
    "detail-type": "Inspector2 Scan",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T22:52:35Z",
    "region": "us-east-1",
    "resources": [
        "i-087d63509b8c97098"
    ],
    "detail": {
        "scan-status": "INITIAL_SCAN_COMPLETE",
        "finding-severity-counts": {
            "CRITICAL": 0,
            "HIGH": 0,
            "MEDIUM": 0,
            "TOTAL": 0
        },
        "instance-id": "i-087d63509b8c97098",
```

```
        "version": "1.0"
    }
}
```

Amazon ECR image initial scan

```
{
    "version": "0",
    "id": "fdaa751a-984c-a709-44f9-9a9da9cd3606",
    "detail-type": "Inspector2 Scan",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T23:15:18Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ecr:us-east-1:111122223333:repository/inspector2"
    ],
    "detail": {
        "scan-status": "INITIAL_SCAN_COMPLETE",
        "repository-name": "arn:aws:ecr:us-east-1:111122223333:repository/
inspector2",
        "finding-severity-counts": {
            "CRITICAL": 0,
            "HIGH": 0,
            "MEDIUM": 0,
            "TOTAL": 0
        },
        "image-digest":
 "sha256:965fbcae990b0467ed5657caceaec165018ef44a4d2d46c7cdea80a9dff0d1ea",
        "image-tags": [
            "ubuntu22"
        ],
        "version": "1.0"
    }
}
```

Lambda function initial scan

```
{
  "version": "0",
  "id": "4f290a7c-361b-c442-03c8-a629f6f20d6c",
  "detail-type": "Inspector2 Scan",
  "source": "aws.inspector2",
  "account": "111122223333",
  "time": "2023-02-23T18:06:03Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:lambda:us-west-2:111122223333:function:lambda-example:$LATEST"
  ],
  "detail": {
    "scan-status": "INITIAL_SCAN_COMPLETE",
    "finding-severity-counts": {
      "CRITICAL": 0,
      "HIGH": 0,
      "MEDIUM": 0,
      "TOTAL": 0
    },
    "version": "1.0"
  }
}
```

Beispiel für ein Amazon Inspector Inspector-Abdeckungsereignisschema

Im Folgenden finden Sie ein Beispiel für das EventBridge Ereignisschema für ein Amazon Inspector Inspector-Ereignis zur Berichterstattung. Dieses Ereignis wird ausgelöst, wenn die Scanabdeckung von Amazon Inspector für eine Ressource geändert wird. Die folgenden Felder kennzeichnen ein Abdeckungsereignis:

- Das `detail-type` Feld ist auf eingestelltInspector2 Coverage.
- Das `detail` Objekt enthält ein `scanStatus` Objekt, das den neuen Scanstatus für die Ressource angibt.

Das `detail` Objekt umfasst auch die folgenden Felder. Diese Felder sind für alle Ressourcen, einschließlich AWS Ressourcen, vorhanden.

- `provider`ist der Cloud-Anbieter der Ressource, zum Beispiel `AWS` oder`AZURE`.
- `providerAccountId`ist die Konto-ID des Cloud-Anbieters für die Ressource, z. B. eine Azure-Abonnement-ID. Bei AWS Ressourcen entspricht dies dem `account` Feld.
- `providerRegion`ist die Cloud-Provider-Region der Ressource. Bei AWS Ressourcen entspricht dies dem `region` Feld.
- `providerOrgId`ist die Organisations- oder Mandanten-ID des Cloud-Anbieters für die Ressource, z. B. eine Azure-Mandanten-ID. Amazon Inspector füllt dieses Feld für `AZURE` Ressourcen aus.

Wählen Sie eine der Optionen aus, um sich je nach Ressourcentyp unterschiedliche Schemas für die Berichterstattung anzeigen zu lassen.

Amazon EC2 instance coverage

```
{
    "version": "0",
    "id": "000adda5-0fbf-913e-bc0e-10f0376412aa",
    "detail-type": "Inspector2 Coverage",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T22:51:39Z",
    "region": "us-east-1",
    "resources": [
        "i-087d63509b8c97098"
    ],
    "detail": {
        "scanStatus": {
            "reason": "UNMANAGED_EC2_INSTANCE",
            "statusCodeValue": "INACTIVE"
        },
        "scanType": "PACKAGE",
        "provider": "AWS",
        "providerAccountId": "111122223333",
        "providerRegion": "us-east-1",
        "eventTimestamp": "2023-01-20T22:51:35.665501Z",
        "version": "1.0"
    }
}
```

Azure virtual machine coverage

```
{
    "version": "0",
    "id": "111b1dda-0fbf-913e-bc0e-10f0376412bb",
    "detail-type": "Inspector2 Coverage",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T22:51:39Z",
    "region": "us-east-1",
    "resources": [
        "/subscriptions/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee/resourcegroups/example-
rg/providers/microsoft.compute/virtualmachines/example-vm"
    ],
    "detail": {
        "scanStatus": {
            "reason": "SUCCESSFUL",
            "statusCodeValue": "ACTIVE"
        },
        "scanType": "PACKAGE",
        "provider": "AZURE",
        "providerAccountId": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
        "providerRegion": "eastus",
        "providerOrgId": "11111111-2222-3333-4444-555555555555",
        "eventTimestamp": "2023-01-20T22:51:35.665501Z",
        "version": "1.0"
    }
}
```

Beispiel für ein Schema zur auto Aktivierung von Amazon Inspector

Das automatische Aktivierungsereignis wird an den delegierten Administrator gesendet, wenn Amazon Inspector die Anzahl der Mitglieder in einer Organisation nicht unterstützen kann. Die folgenden Felder identifizieren ein automatisches Aktivierungsereignis:

- Das `detail-type` Feld ist auf eingestellt. `Inspector2 AutoEnable`
- Das `detail` Objekt beschreibt, warum das auto Aktivierungsereignis fehlgeschlagen ist.

```
{
    "version": "0",
    "id": "85fc3613-e913-7fc4-a80c-a3753e4aa9ae",
    "detail-type": "Inspector2 AutoEnable",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2024-08-21T02:36:48Z",
    "region": "us-east-1",
    "detail": {
        “version”: “1.0.0”,
        “AutoEnableStatus”: “Failed”,
        “Reason”: "The number of member accounts enabled with AWS Inspector has reached
 the maximum limit of 10,000"
        }
}
```

Amazon Inspector VM-Scanner

-Übersicht

Amazon Inspector verwendet den Amazon Inspector VM Scanner, um Sicherheitslücken auf Amazon EC2 EC2-Instances zu bewerten. Amazon Inspector VM Scanner nutzt die Inventarerfassungsfunktionen von Amazon Inspector SBOM Generator, um eine Software-Stückliste (SBOM) zu erstellen, und reicht die SBOM zur Auswertung über den Inspector Telemetrie-Kanal ein. Wenn Sie Enhanced EC2 Scanning für Ihr Konto aktivieren, ersetzt Inspector VM Scanner das Inspector SSM Plugin.

> **Empfehlung**
>
> Wir empfehlen den Amazon Inspector VM Scanner dem älteren Amazon Inspector SSM-Plugin vorzuziehen. Der VM-Scanner verwendet für alle unterstützten Betriebssysteme den gleichen Scanmechanismus und liefert konsistentere Ergebnisse. Außerdem benötigt er weniger Rechenressourcen. WindowsInsbesondere bei werden Timeouts pro Abfrage vermieden, die dazu führen können, dass das Amazon Inspector SSM-Plugin Ergebnisse inkonsistent meldet.

Amazon Inspector bietet zwei Ansätze für die Bereitstellung des VM-Scanners:

- Automatische Installation (empfohlen) — Wenn Sie Enhanced EC2 Scanning in der Amazon Inspector-Konsole aktivieren, verwendet Amazon Inspector Amazon EC2 Systems Manager (SSM), um den VM-Scanner automatisch auf Ihren Amazon EC2-Instances zu installieren und zu verwalten. Dies ist der einfachste Ansatz und erfordert kein manuelles Eingreifen. Weitere Informationen finden Sie unter Inspector VM Scanner aktivieren.
- Manuelle Installation — Sie können den VM-Scanner mithilfe von Standard-Paketmanagern (RPM, DEB, APK, MSI, PKG) manuell installieren. Für diesen Ansatz ist kein SSM erforderlich. Weitere Informationen finden Sie unter Manuelle Installation und Konfiguration.

Vorteile

Inspector VM Scanner bietet die folgenden Vorteile gegenüber dem vorherigen Inspector SSM Plugin:

- Reduzierter Ressourcenverbrauch — Erfordert weniger Rechenleistung während des Betriebs
- Granularere Paketerfassung — Bietet ein detailliertes Inventar auf Paketebene für eine genauere Bewertung
- Verbesserter Scanmechanismus — Verwendet Inspector SBOM Generator für ein einheitliches Verhalten mit anderen von Inspector unterstützten Ressourcen
- Aufgabentrennung — Sicherheitsteams können das Scannen auf Kontoebene ermöglichen, während die Instanzadministratoren die Kontrolle über die manuelle Installation und Konfiguration der einzelnen Instanzen behalten

Gründliche Inspektion

> **Verbesserte Scanunterstützung**
>
> Verbessertes Scannen ermöglicht tiefe Inspektionspfade bei WindowsLinux, undmacOS.

Mit Enhanced EC2 Scanning erweitert der Amazon Inspector VM Scanner die Reichweite von Amazon EC2-Scans um eine gründliche Inspektion. Mit einer gründlichen Prüfung erkennt Amazon Inspector Sicherheitslücken in Paketen für Anwendungsprogrammiersprachen sowie in Betriebssystempaketen. Amazon Inspector scannt Standardspeicherorte nach Paketbibliotheken für Programmiersprachen, und Sie können zusätzlich zu den Speicherorten, die Amazon Inspector standardmäßig scannt, benutzerdefinierte Pfade konfigurieren.

> **Important**
>
> Amazon Inspector scannt eine Reihe von Standardspeicherorten, die je nach Betriebssystem variieren. Software an einem nicht standardmäßigen Speicherort — wie z. B. einem Datenbank- oder Anwendungsserver auf einem Laufwerk `D:\` — wird nur gescannt, wenn Sie sie als benutzerdefinierten Pfad hinzufügen. Weitere Informationen finden Sie unter Benutzerdefinierte Pfade für Amazon Inspector Deep Inspection.

Informationen zu den Programmiersprachen, die Deep Inspection unterstützt, finden Sie unter Unterstützte Programmiersprachen: Amazon EC2 Deep Inspection.

Inspector VM Scanner aktivieren

Wenn Sie Enhanced EC2 Scanning in der Amazon Inspector-Konsole aktivieren, verwendet Amazon Inspector Amazon EC2 Systems Manager (SSM), um den VM-Scanner automatisch auf Ihren verwalteten Amazon EC2-Instances zu installieren. Nach der Installation wird der Scanner regelmäßig (standardmäßig alle 3 Stunden) ausgeführt und sendet Ergebnisse an den Amazon Inspector Telemetry Service.

Voraussetzungen

Um die automatische Installationsmethode verwenden zu können, müssen Ihre Amazon EC2 EC2-Instances die folgenden Anforderungen erfüllen:

- Der SSM-Agent muss auf der Instance installiert sein und ausgeführt werden. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent im AWS Systems Manager - Benutzerhandbuch.
- Die Instanz muss über ein IAM-Instanzprofil verfügen, das es SSM ermöglicht, die Instanz zu verwalten. Weitere Informationen finden Sie im AWS Systems Manager Benutzerhandbuch unter Instanzberechtigungen für Systems Manager konfigurieren.
- Die Instanz muss über eine Netzwerkverbindung zu den SSM-Dienstendpunkten verfügen.

> **Note**
>
> Wenn auf Ihren Instanzen kein SSM-Agent installiert ist oder diese Anforderungen nicht erfüllt werden können, können Sie stattdessen die manuelle Installationsmethode verwenden. Weitere Informationen finden Sie unter Manuelle Installation und Konfiguration.

Enhanced EC2 Scanning aktivieren

Um das erweiterte EC2-Scannen zu aktivieren und den VM-Scanner automatisch zu installieren:

1. Öffnen Sie die Amazon Inspector Inspector-Konsole unter Getting Started mit dem AWS-Managementkonsole.
2. Wählen Sie im Navigationsbereich Kontoverwaltung aus.
3. Wählen Sie unter EC2-Scannen die Option Bearbeiten aus.
4. Aktivieren Sie das erweiterte EC2-Scannen.

Nachdem Sie Enhanced EC2 Scanning aktiviert haben, erstellt Amazon Inspector eine SSM-Zuordnung, die den VM-Scanner auf allen geeigneten Instances in Ihrem Konto installiert. Der Scanner beginnt automatisch mit der Ausführung von Schwachstellenanalysen.

> **Note**
>
> Warnung: Wenn Sie Enhanced EC2 Scanning mit dem Inspector VM Scanner aktivieren, wird das Inspector SSM-Plugin dauerhaft deaktiviert. Dies gilt sowohl für Einzelpersonen AWS-Konten als auch für Organisationen, die der delegierte Administrator aktiviert. Nachdem Sie diese Funktion aktiviert haben, können Sie nicht mehr zum Inspector SSM-Plugin zurückkehren.

> **Note**
>
> Bevor Sie Enhanced EC2 Scanning aktivieren, sollten Sie in Erwägung ziehen, zunächst in einem Konto zu testen, das nicht für die Produktion bestimmt ist. Wenn Sie nicht zuerst testen, können Sie nicht zum Inspector SSM-Plugin zurückkehren, falls Probleme auftreten. Weitere Informationen finden Sie unter Manuelle Installation und Konfiguration.

Amazon VPC-Endpunktanforderungen für erweitertes EC2-Scannen auf privaten Amazon EC2 EC2-Instances

Sie können Enhanced EC2 Scanning auf Amazon EC2 EC2-Instances über ein Amazon-Netzwerk ausführen. Wenn Sie jedoch Enhanced EC2 Scanning auf privaten Amazon EC2 EC2-Instances ausführen möchten, müssen Sie Amazon VPC-Endpoints erstellen. Die folgenden Endpunkte sind erforderlich:

- `com.amazonaws.region.inspector2-telemetry`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ssm`
- `com.amazonaws.region.ssmmessages`

Wo *region* ist der Regionalcode für den AWS-Region zutreffenden.

Weitere Informationen finden Sie unter Verbessern der Sicherheit von Amazon EC2 EC2-Instances mithilfe von Amazon VPC-Endpunkten für Systems Manager im AWS Systems Manager Benutzerhandbuch.

> **Note**
>
> Ab SSM Agent Version 3.3.40.0 verwendet SSM Agent den `ssmmessages` Endpunkt anstelle des Endpunkts, wann immer dieser verfügbar ist. `ec2messages` Frühere Versionen von SSM Agent verwenden den Endpunkt. `ec2messages` Dieser Endpunkt wird nur unterstützt, wenn er vor 2024 AWS-Regionen gestartet wurde. Weitere Informationen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen im Benutzerhandbuch.AWS Systems Manager

Häufig gestellte Fragen

Warum sollte ich zu Inspector VM Scanner migrieren? Wie unterscheidet sich Inspector VM Scanner vom Inspector SSM Plugin?

Inspector VM Scanner verwendet Amazon Inspector SBOM Generator zum Scannen von Systempaketen und zur Tiefeninspektion und bietet so einen konsistenten Scanmechanismus, unabhängig davon, wie Pakete installiert wurden. Dieser Scanmechanismus ist auch leistungsfähiger, wobei der größte Leistungszuwachs zu verzeichnen ist. Windows Dieser Mechanismus gehört vollständig Inspector und wird von ihm verwaltet, sodass Inspector schnell auf auftretende Probleme reagieren und die Erkennungen auf neue Ökosysteme ausweiten kann.

Wie unterscheidet sich Inspector VM Scanner von Inspector SBOM Generator? Warum nicht SBOM Generator direkt verwenden?

Inspector VM Scanner verwendet den SBOM Generator unter der Haube für die Inventarerfassung, enthält aber auch zusätzliche Funktionen, die speziell für Inspector EC2 Scanning entwickelt wurden. Inspector VM Scanner kann über den Inspector Telemetrie-Kanal kommunizieren, Inspector SBOM Generator dagegen nicht. Darüber hinaus verfügt Inspector VM Scanner über eine Orchestrierung, die regelmäßig für geplante Scans aufgerufen werden kann.

Wie werden die Scans des Center for Internet Security (CIS) verwaltet? Wird Inspector VM Scanner für CIS verwendet?

CIS-Scans werden im Inspector SSM Plugin gespeichert. Derzeit ist nicht beabsichtigt, die CIS-Unterstützung auf Inspector VM Scanner zu migrieren.

Benötigt Inspector VM Scanner SSM?

Nein. SSM wird für die automatische Installation verwendet, wenn Sie Enhanced EC2 Scanning in der Konsole aktivieren, aber es handelt sich nicht um eine feste Abhängigkeit. Sie können den VM-Scanner manuell mithilfe von Standard-Paketmanagern ohne SSM installieren. Weitere Informationen finden Sie unter Manuelle Installation und Konfiguration.

Scannt Inspector VM Scanner dieselben Standorte wie das Inspector SSM Plugin?

Nicht unbedingt. Inspector VM Scanner scannt einen Standardsatz von Speicherorten, der je nach Betriebssystem variiert und sich vom Inspector SSM Plugin unterscheiden kann. Software, die an einem benutzerdefinierten Speicherort installiert ist, wird nur gescannt, wenn Sie sie als benutzerdefinierten Pfad hinzufügen. Mit Enhanced EC2 Scanning können Sie benutzerdefinierte Pfade für LinuxWindows, und macOS Instances festlegen. Wenn Sie Amazon Inspector aktivieren oder erneut aktivieren, überprüfen Sie Ihre benutzerdefinierten Pfade, bevor Sie sich auf die Ergebnisse verlassen. Weitere Informationen finden Sie unter Benutzerdefinierte Pfade für Amazon Inspector Deep Inspection.

Third-party Softwarezuweisung

Inspector VM Scanner enthält Softwarekomponenten von Drittanbietern, die unter Open-Source-Lizenzen lizenziert sind. Gemäß den Bedingungen dieser Lizenzen werden im folgenden Dokument zur Quellenangabe die Komponenten von Drittanbietern, ihre Lizenzen und der entsprechende Lizenztext aufgeführt.

```
https://inspector-vm-scanner.s3.amazonaws.com/latest/THIRD-PARTY-LICENSES.txt
```

Manuelle Installation und Konfiguration

In diesem Abschnitt wird beschrieben, wie Sie Inspector VM Scanner manuell auf Ihren Amazon EC2 EC2-Instances installieren und konfigurieren. Die manuelle Installation gilt als agentengestütztes Scannen und erfordert keinen Amazon EC2 Systems Manager (SSM).

> **Note**
>
> Wenn Sie Enhanced EC2 Scanning in der Amazon Inspector-Konsole aktivieren, installiert und verwaltet Amazon Inspector den VM-Scanner automatisch mithilfe von SSM. Eine

manuelle Installation ist nur erforderlich, wenn Sie SSM nicht verwenden können oder den Scanner lieber unabhängig verwalten möchten.

Manuelles Installieren und Deinstallieren von Inspector VM Scanner

Inspector VM Scanner ist für die eigenständige Installation über die folgenden Links verfügbar:

Inspector VM Scanner-Pakete

Art Package	Architektur	URL
.apk	ARM	https://inspector-vm-scanner.s3.amazonaws.com/latest/APK-ARM/inspector-vm-scanner-latest-r0.apk
	X86_64	https://inspector-vm-scanner.s3.amazonaws.com/latest/APK-X86_64/inspector-vm-scanner-latest-r0.apk
.deb	ARM	https://inspector-vm-scanner.s3.amazonaws.com/latest/DEB-ARM/inspector-vm-scanner_latest_arm64.deb
	X86_64	https://inspector-vm-scanner.s3.amazonaws.com/latest/DEB-X86_64/inspector-vm-scanner_latest_amd64.deb
.msi	X86_64	https://inspector-vm-scanner.s3.amazonaws.com/latest/MSI-X86_64/inspector-vm-scanner-x86_64-latest.msi
.pkg	ARM	https://inspector-vm-scanner.s3.amazonaws.com/latest/PKG-ARM/inspector-vm-scanner.latest.arm64.pkg
.rpm	ARM	https://inspector-vm-scanner.s3.amazonaws.com/latest/RPM-ARM/inspector-vm-scanner-latest.arm64.rpm
	X86_64	https://inspector-vm-scanner.s3.amazonaws.com/latest/RPM-X86_64/inspector-vm-scanner-latest.x86_64.rpm

Um das Verfahren zur manuellen Installation von Inspector VM Scanner auf dem angegebenen Paketmanager anzuzeigen, wählen Sie einen Link aus der folgenden Liste:

- APK

- DEB
- MSI
- PKG
- RPM

APK

Installation

ARM

```
curl --output-dir /etc/apk/keys -O https://inspector-vm-scanner.s3.amazonaws.com/
latest/APK-ARM/inspector-vm-scanner.pem.pub
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/APK-ARM/inspector-vm-
scanner-latest-r0.apk
apk add inspector-vm-scanner-latest-r0.apk
```

X86_64

```
curl --output-dir /etc/apk/keys -O https://inspector-vm-scanner.s3.amazonaws.com/
latest/APK-X86_64/inspector-vm-scanner.pem.pub
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/APK-X86_64/inspector-vm-
scanner-latest-r0.apk
apk add inspector-vm-scanner-latest-r0.apk
```

Deinstallation

```
apk del inspector-vm-scanner
```

DEB

Installation

ARM

```
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/DEB-ARM/inspector-vm-
scanner.gpg.pub
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/DEB-ARM/inspector-vm-
scanner_latest_arm64.deb
```

```
gpg --import inspector-vm-scanner.gpg.pub
gpg --verify inspector-vm-scanner_latest_arm64.deb
sudo dpkg -i inspector-vm-scanner_latest_arm64.deb
```

X86_64

```
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/DEB-X86_64/inspector-vm-scanner.gpg.pub
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/DEB-X86_64/inspector-vm-scanner_latest_amd64.deb
gpg --import inspector-vm-scanner.gpg.pub
gpg --verify inspector-vm-scanner_latest_amd64.deb
sudo dpkg -i inspector-vm-scanner_latest_amd64.deb
```

Deinstallation

```
sudo dpkg -r inspector-vm-scanner
```

MSI

Installation

X86_64

```
Invoke-WebRequest https://inspector-vm-scanner.s3.amazonaws.com/latest/MSI-X86_64/inspector-vm-scanner-x86_64-latest.msi -OutFile inspector-vm-scanner-x86_64-latest.msi
msiexec /i inspector-vm-scanner-x86_64-latest.msi /qn
```

Deinstallation

Um Inspector VM Scanner auf zu deinstallierenWindows, verwenden Sie das Bedienfeld Windows Programme und Funktionen oder den folgenden PowerShell Befehl:

```
Get-WmiObject -Class Win32_Product | Where-Object {$_.Name -eq "Inspector VM Scanner"}
 | ForEach-Object {$_.Uninstall()}
```

PKG

Installation

ARM

```
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/PKG-ARM/inspector-vm-
scanner.latest.arm64.pkg
pkgutil --check-signature inspector-vm-scanner.latest.arm64.pkg
sudo installer -pkg inspector-vm-scanner.latest.arm64.pkg -target /
```

Deinstallation

```
sudo rm /opt/aws/inspector/bin/inspector-vm-scanner
sudo rm -rf /var/lib/amazon/inspector
```

RPM

Installation

ARM

```
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/RPM-ARM/inspector-vm-
scanner.gpg.pub
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/RPM-ARM/inspector-vm-
scanner-latest.arm64.rpm
rpm --import inspector-vm-scanner.gpg.pub
rpm --checksig inspector-vm-scanner-latest.arm64.rpm
sudo yum install inspector-vm-scanner-latest.arm64.rpm
```

X86_64

```
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/RPM-X86_64/inspector-vm-
scanner.gpg.pub
curl -O https://inspector-vm-scanner.s3.amazonaws.com/latest/RPM-X86_64/inspector-vm-
scanner-latest.x86_64.rpm
rpm --import inspector-vm-scanner.gpg.pub
rpm --checksig inspector-vm-scanner-latest.x86_64.rpm
sudo yum install inspector-vm-scanner-latest.x86_64.rpm
```

Deinstallation

```
sudo yum remove inspector-vm-scanner
```

Installationspfade

Auf allen Unix-based Betriebssystemen (einschließlich macOS) wird Inspector VM Scanner unter installiert`/opt/aws/inspector/bin/inspector-vm-scanner`. Die Ausnahme bilden Alpine-based Betriebssysteme (einschließlich Chainguard), die den alternativen Pfad verwenden. `/usr/opt/aws/inspector/bin/inspector-vm-scanner`

EinWindows, Inspector VM Scanner wird auf installiert`C:\Program Files\Amazon\Inspector\inspector-vm-scanner.exe`.

Diese Installationspfade (mit Ausnahme von Alpine) sind identisch mit dem Inspector SSM Plugin, das alle Inspector-Binärdateien an einem einzigen Ort speichert.

Deinstallation von Inspector VM Scanner

Wenn Sie Enhanced EC2 Scanning deaktivieren, deinstalliert Inspector VM Scanner automatisch auf allen Inspector-managed Instanzen. Informationen zum Entfernen einer manuellen Installation finden Sie in den Anweisungen zur Deinstallation Ihres Paketmanagers in den vorherigen Abschnitten.

Inspector VM Scanner ausführen

Inspector VM Scanner erwartet, dass ein Scan-Typ als erster Parameter übergeben wird. Derzeit ist der einzige unterstützte Wert`sbom`.

Standard-Verwendungsbefehl:

```
./inspector-vm-scanner sbom --send-results telemetry
```

Druckoptionen für den SBOM-Scan:

```
./inspector-vm-scanner sbom --help
```

Ausgabe anzeigen

Der standardmäßige Inspector-Workflow speichert keine SBOM lokal. Wenn beim Senden der Ressourcen-SBOM über Telemetrie jedoch Fehler auftreten, wird sie an die folgenden Speicherorte geschrieben:

- `/var/lib/amazon/inspector/state/sbom.json`unter Unix

- `C:\ProgramData\Amazon\Inspector\State\sbom.json` auf Windows

Benutzer können diesen Pfad beim Aufruf des VM-Scanners überschreiben. Weitere Details finden Sie unter Fortgeschrittene Konfiguration.

Anzeigen von Protokollen

Inspector VM Scanner-Protokolle beginnen immer mit `inspector-vm-scanner` und enden mit `.log`.

Auf Unix-Systemen schreibt Inspector VM Scanner Protokolle in `/var/log/amazon/inspector`. Ist aktiviertWindows, schreibt Inspector VM Scanner Protokolle in `C:\ProgramData\Amazon\Inspector\Logs`.

Beide Pfade sind identisch mit dem, was das Inspector SSM Plugin verwendet hat. Dies bedeutet, dass Sie möglicherweise alte Inspector SSM-Plug-in-Protokolle neben Inspector VM Scanner-Protokollen sehen.

Inspector VM Scanner kann auch optionale Metrikprotokolle ausgeben, die die CPU- und Speicherauslastung während der Programmausführung verfolgen. Diese Protokolle werden in die zuvor genannten Protokollverzeichnisse in einem Unterverzeichnis mit dem Namen `metrics` (`Metrics`onWindows) geschrieben. Weitere Details finden Sie unter Fortgeschrittene Konfiguration.

Logs mit Systemd anzeigen

Auf Linux-Systemen, die systemd verwenden, können Sie journalctl damit Inspector VM Scanner-Protokolle anzeigen.

Protokolle in Echtzeit anzeigen

```
journalctl -u inspector-vm-scanner -f
```

Aktuelle Protokolle anzeigen

```
journalctl -u inspector-vm-scanner --since "1 hour ago"
```

Logs von heute anzeigen

```
journalctl -u inspector-vm-scanner --since today
```

Den Dienststatus und die letzten Läufe anzeigen

```
systemctl status inspector-vm-scanner
```

Fortgeschrittene Konfiguration

In diesem Abschnitt werden die erweiterten Konfigurationsoptionen für Inspector VM Scanner beschrieben.

Konfiguration lokaler Ausgaben

Inspector VM Scanner bietet die folgenden Optionen, um zu konfigurieren, wie lokale Ausgaben geschrieben werden:

- `--send-results`muss auf `telemetry` oder gesetzt werden`disabled`. Wenn Sie die Prüfung bestehen`disabled`, fährt Inspector VM Scanner fort, ohne die SBOM zu senden.

> **Tip**
>
> Verwenden Sie `--state-dir` with`--send-results disabled`, um die SBOM lokal zu speichern.

- `--log-dir`konfiguriert, wo Protokolle geschrieben werden. Standardmäßig werden Protokolle auf stdout geschrieben.
- `--log-level`konfiguriert die Granularität von Protokollen. In der Standardeinstellung ist dies INFO.
- `--log-retention`konfiguriert, wie viele Tage Protokolle aufbewahrt werden sollen. Wenn eine Protokolldatei älter `--log-retention` ist als in`--log-dir`, wird sie gelöscht. In der Standardeinstellung sind dies 7 Tage.
- `--debug`konfiguriert die Protokollierung auf Debug-Ebene und erzwingt eine dedizierte Protokolldatei für die aktuelle Ausführung (anstatt zu versuchen, für jeden Tag eine Protokolldatei zu verwalten).
- `--state-dir`konfiguriert, wo SBOMs geschrieben werden. Standardmäßig werden SBOMs nicht gespeichert.

- `--metric-dir`konfiguriert, wo Metrikprotokolle geschrieben werden. Standardmäßig werden Metrikprotokolle nicht gespeichert.
- `--cpu-profile`aktiviert den Go-Laufzeit-CPU-Profiler und konfiguriert, wo das Ergebnis geschrieben wird.
- `--mem-profile`aktiviert den Go-Laufzeitspeicherprofiler und konfiguriert, wo das Ergebnis geschrieben wird.
- `--config-path`weist Inspector VM Scanner an, Argumente aus einer lokalen Konfigurationsdatei abzuleiten. Wenn dasselbe Argument sowohl in der CLI als auch in der Konfigurationsdatei übergeben wird, wird der CLI-Wert priorisiert.
  - Die Konfigurationsdateien von Inspector VM Scanner werden in TOML angegeben, wobei alle Argumentnamen mit denen der CLI identisch sind.
  - Standardmäßig sucht Inspector VM Scanner unter Linux und macOS `/opt/aws/inspector/etc/config` nach einer Konfigurationsdatei oder `C:\\ProgramData\\Amazon\\Inspector\\Config\\config` unterWindows. Die Datei hat keine Erweiterung, verwendet jedoch das TOML-Format.
  - Wenn Sie angeben`--config-path`, hat der angegebene Pfad Vorrang vor dem Standardpfad.

Das folgende Beispiel zeigt eine Konfigurationsdatei:

```
# Configuration file for Inspector VM Scanner
log-level = "INFO"
send-results = "telemetry"
cpu-profile = "cpuprofile"
mem-profile = "memprofile"
log-dir = "log"
state-dir = "state"
debug = false
log-retention = 7
scan-timeout = 300

[sbom]
max-scan-depth = 5
target-directory = ["~"]
```

Konfiguration der Ressourcennutzung

Inspector VM Scanner bietet die folgenden Optionen zur Konfiguration der Ressourcennutzung:

- `--scan-timeout`zwingt den Scanner nach einer bestimmten Anzahl von Sekunden zu einem Timeout. Standardmäßig läuft beim Scanner kein Timeout ab.
- `--nice-priority`legt die `nice` Priorität für den Prozess fest (verfügbar für Unix-Systeme). In der Standardeinstellung ist dies 3.
- `--cpu-limit`legt eine feste Obergrenze für die CPU-Auslastung fest (verfügbar für Linux-Systeme, die diese verwenden`cgroups`). In der Standardeinstellung sind dies 65%.
- `--process-priority`konfiguriert die Priorität für den Prozess (verfügbar für Windows Systeme). Standardmäßig ist dies die `BELOW NORMAL` Priorität.

> **Note**
>
> Die Standardwerte für `--cpu-limit` und `--process-priority` sind identisch mit denen des Inspector SSM Plugins.

Scanziele konfigurieren

Inspector VM Scanner nutzt Inspector SBOM Generator für die Inventarerfassung. Daher werden viele der Scan-Abdeckungsoptionen von Inspector VM Scanner direkt aus dem SBOM Generator übernommen.

Standardmäßig scannt Inspector VM Scanner nicht Ihr gesamtes Dateisystem. Es sammelt Betriebssystempakete und verwendet Windows dann Windows Update-Daten (Knowledge Base), um Systemsoftware zu identifizieren, die aktualisiert werden muss. Außerdem werden gängige Installationsorte für beliebte Programmiersprachenpakete und deren Abhängigkeiten untersucht.

Software, die an einem benutzerdefinierten oder nicht standardmäßigen Speicherort installiert ist — z. B. in einer Datenbank oder einem Anwendungsserver auf einem System-Laufwerk — wird nur gescannt, wenn Sie sie hinzufügen. Um solche Software abzudecken, fügen Sie sie als benutzerdefinierten Pfad in Ihren Amazon EC2-Scaneinstellungen hinzu. Unter Enhanced EC2 Scanning gelten benutzerdefinierte Pfade für LinuxWindows, und macOS Instances und werden zusätzlich zu den Standardspeicherorten gescannt. Weitere Informationen finden Sie unter Benutzerdefinierte Pfade für Amazon Inspector Deep Inspection.

Mithilfe der erweiterten Optionen in diesem Abschnitt können Sie beim Aufruf des VM-Scanners auch Verzeichnisse pro Instanz hinzufügen. Verzeichnisse, die Sie hinzufügen, `--target-directory` werden zusätzlich zu den Standardwerten gescannt.

Inspector VM Scanner bietet die folgenden Optionen zur Konfiguration von Scanzielen:

- `--max-scan-depth`konfiguriert die maximale Anzahl von Verzeichnisebenen, die ein Scan von jedem Scan-Standort aus durchläuft.
- `--target-directory`fügt zusätzlich zu den Standardwerten ein oder mehrere Verzeichnisse hinzu, die gescannt werden sollen. Geben Sie es in einer Konfigurationsdatei unter beispielsweise `[sbom] target-directory = ["D:\\oracle"]` als Array an.
- `--override-scanners`konfiguriert exakte Dateiscanner und überschreibt die Standardeinstellungen von Inspector VM Scanner.
- `--additional-scanners`konfiguriert Dateiscanner, die zusätzlich zu den Standardeinstellungen von Inspector VM Scanner verwendet werden.

Sie können den folgenden Befehl verwenden, um alle verfügbaren Scanner aufzulisten:

```
./inspector-vm-scanner sbom --list-scanners
```

Verwaltung der regelmäßigen Ausführung

Wenn Sie Inspector VM Scanner über einen Paketmanager installieren, erstellt die Installation eine geplante Aufgabe, die Scans automatisch ausführt. Sie können diesen Zeitplan anzeigen, ändern oder deaktivieren.

Linux (systemd)

Den Dienststatus und die letzten Läufe anzeigen

```
systemctl status inspector-vm-scanner
```

Protokolle in Echtzeit anzeigen

```
journalctl -u inspector-vm-scanner -f
```

Aktuelle Protokolle anzeigen

```
journalctl -u inspector-vm-scanner --since "1 hour ago"
```

Überprüfen Sie das aktuelle Timer-Intervall

```
systemctl cat inspector-vm-scanner.timer
```

Ändern Sie das Scanintervall

> **Important**
>
> Bearbeiten Sie die Unit-Dateien nicht `/usr/lib/systemd/system/` direkt. Paket-Updates überschreiben diese Dateien. Verwenden Sie `systemctl edit` stattdessen diese Option. Dadurch wird eine persistente Überschreibung erstellt, die auch nach Aktualisierungen gültig ist.

Das Standardintervall ist alle 3 Stunden. Um es zu ändern, führen Sie Folgendes aus:

```
sudo systemctl edit inspector-vm-scanner.timer
```

Fügen Sie im sich öffnenden Editor Folgendes hinzu (in diesem Beispiel wird das Intervall auf 6 Stunden festgelegt):

```
[Timer]
OnBootSec=
OnUnitActiveSec=
OnBootSec=0
OnUnitActiveSec=21600s
```

Die leeren Zuweisungen löschen die Standardwerte, bevor Ihre Werte wirksam werden. Starten Sie den Timer nach dem Speichern neu:

```
sudo systemctl daemon-reload
sudo systemctl restart inspector-vm-scanner.timer
```

Führen Sie Folgendes aus, um zu überprüfen, ob die Überschreibung wirksam wurde`systemctl cat inspector-vm-scanner.timer`. Um zu den Standardwerten zurückzukehren, führen Sie den Befehl aus. `sudo systemctl revert inspector-vm-scanner.timer`

Aktiviere oder deaktiviere die automatische Ausführung

```
sudo systemctl enable inspector-vm-scanner.timer   # Enable automatic runs.
sudo systemctl disable inspector-vm-scanner.timer  # Disable automatic runs.
```

Windows (Taskplaner)

Den Aufgabenstatus und die letzte Ausführung anzeigen

```
Get-ScheduledTask -TaskName "Inspector VM Scanner" | Get-ScheduledTaskInfo
```

Aktuelle Aufgabenprotokolle anzeigen

```
Get-ScheduledTaskInfo -TaskName "Inspector VM Scanner"
```

Detaillierten Aufgabenverlauf anzeigen

```
schtasks /query /tn "Inspector VM Scanner" /v /fo list
```

Den aktuellen Aufgabenplan anzeigen

```
Get-ScheduledTask -TaskName "Inspector VM Scanner" | Select-Object -ExpandProperty
 Triggers
```

Aufgabenplan aktualisieren

Um die Scanfrequenz zu ändern:

```
# Modify trigger to run daily at 2 AM
$trigger = New-ScheduledTaskTrigger -Daily -At 2:00AM
Set-ScheduledTask -TaskName "Inspector VM Scanner" -Trigger $trigger
```

Aufgabe aktivieren oder deaktivieren

```
Enable-ScheduledTask -TaskName "Inspector VM Scanner"   # Enable automatic runs
Disable-ScheduledTask -TaskName "Inspector VM Scanner"  # Disable automatic runs
```

macOS (gestartet)

Gestartete Aufgaben anzeigen

```
sudo launchctl print system/com.amazon.inspector.vm-scanner
```

Einzelne Aufgabe ausführen

```
sudo launchctl start com.amazon.inspector.vm-scanner
```

Amazon Inspector SSM-Plugin für Linux und Windows

In diesem Thema wird das Amazon Inspector SSM-Plugin für Linux Windows Und-Instances beschrieben.

> ⚠ Wir empfehlen Enhanced EC2 Scanning
>
> Das Amazon Inspector SSM-Plugin ist der Legacy-Scanner. Wir empfehlen Ihnen, ein Upgrade auf Enhanced EC2 Scanning mit dem Amazon Inspector VM Scanner durchzuführen. Der VM-Scanner verwendet auf allen unterstützten Betriebssystemen den gleichen Scanmechanismus und liefert konsistentere Ergebnisse. WindowsInsbesondere bei werden Timeouts pro Abfrage vermieden, die dazu führen können, dass das Amazon Inspector SSM-Plugin Ergebnisse inkonsistent meldet.

Das Amazon Inspector SSM-Plugin für Linux

Wenn Sie Enhanced EC2 Scanning nicht mit dem Amazon Inspector VM Scanner aktiviert haben, verwendet Amazon Inspector das Amazon Inspector SSM-Plugin, um Deep Inspection-Scans auf Linux-Instances durchzuführen. Das Amazon Inspector SSM-Plugin wird automatisch auf Linux-Instances im `/opt/aws/inspector/bin` Verzeichnis installiert. Der Name der ausführbaren Datei lautet`inspectorssmplugin`.

Amazon Inspector verwendet Systems Manager Distributor, um das Plugin auf Ihrer Instance bereitzustellen. Um Deep Inspection-Scans durchführen zu können, müssen Systems Manager Distributor und Amazon Inspector Ihr Amazon EC2 EC2-Instance-Betriebssystem unterstützen. Informationen zu den Betriebssystemen, die Systems Manager Distributor unterstützt, finden Sie im AWS Systems Manager Benutzerhandbuch unter Unterstützte Paketplattformen und Architekturen.

Amazon Inspector erstellt Dateiverzeichnisse, um Daten zu verwalten, die für die eingehende Prüfung mit dem Amazon Inspector SSM-Plugin gesammelt wurden. Zu diesen Dateiverzeichnissen gehören `/opt/aws/inspector/var/input` und`/opt/aws/inspector/var/output`.

In der `packages.txt` Datei `/opt/aws/inspector/var/output` werden die vollständigen Pfade zu Paketen gespeichert, die bei einer Tiefeninspektion entdeckt wurden. Wenn Amazon Inspector

dasselbe Paket mehrmals auf Ihrer Instance erkennt, werden in der `packages.txt` Datei alle Standorte aufgeführt, an denen das Paket gefunden wurde.

Amazon Inspector speichert Protokolle für das Plugin im `/var/log/amazon/inspector` Verzeichnis.

Deinstallation des Amazon Inspector SSM-Plug-ins

Wenn die `inspectorssmplugin` Datei versehentlich gelöscht wird, versucht die SSM-Zuordnung, die `inspectorssmplugin` Datei beim nächsten `InspectorLinuxDistributor-do-not-delete` Scanintervall erneut zu installieren.

Wenn Sie das Amazon EC2-Scannen deaktivieren, wird das Plugin automatisch von allen Linux-Hosts deinstalliert.

Das Amazon Inspector SSM-Plugin für Windows

Wenn Sie Enhanced EC2 Scanning nicht mit dem Amazon Inspector VM Scanner aktiviert haben, ist das Amazon Inspector SSM-Plugin erforderlich, damit Amazon Inspector Ihre Windows Instances scannen kann. Das Amazon Inspector SSM-Plugin wird automatisch auf Ihren Windows Instances in installiert`C:\Program Files\Amazon\Inspector`, und die ausführbare Binärdatei wird benannt`InspectorSsmPlugin.exe`.

Die folgenden Dateispeicherorte werden erstellt, um Daten zu speichern, die das Amazon Inspector SSM-Plugin sammelt:

- `C:\ProgramData\Amazon\Inspector\Input`
- `C:\ProgramData\Amazon\Inspector\Output`
- `C:\ProgramData\Amazon\Inspector\Logs`

> **Note**
>
> Standardmäßig wird das Amazon Inspector SSM-Plugin mit niedrigerer Priorität ausgeführt.

> **Note**
>
> Für das Scannen Windows von Instances sind `ssm:GetParameter` Berechtigungen erforderlich`ssm:PutInventory`. Wenn auf der Instance ein IAM-Instance-Profil konfiguriert

ist, verwendet Amazon Inspector dieses Profil und ignoriert die Rolle Default Host Management Configuration (DHMC). Das Instance-Profil muss diese Berechtigungen enthalten. Wenn kein Instance-Profil festgelegt ist, verwendet Amazon Inspector die konfigurierte Standard-Host-Management-Konfigurationsrolle, die diese Berechtigungen enthalten muss.

Deinstallation des Amazon Inspector SSM-Plug-ins

Wenn die `InspectorSsmPlugin.exe` Datei versehentlich gelöscht wird, installiert die `InspectorDistributor-do-not-delete` Assoziation die `InspectorSsmPlugin.exe` Datei beim nächsten Scanintervall erneut. Windows Wenn Sie das Amazon Inspector SSM-Plugin deinstallieren möchten, können Sie die Aktion Deinstallieren im `AmazonInspector2-ConfigureInspectorSsmPlugin` Dokument verwenden. Das Amazon Inspector SSM-Plugin wird jedoch automatisch von allen Windows Hosts deinstalliert, wenn Sie das Amazon EC2-Scannen deaktivieren.

> **Note**
>
> Wenn Sie den SSM-Agenten deinstallieren, bevor Sie Amazon Inspector deaktivieren, verbleibt das Amazon Inspector SSM-Plugin auf dem Windows Host, sendet jedoch keine Daten an das Amazon Inspector SSM-Plugin. Weitere Informationen finden Sie unter Amazon Inspector deaktivieren.

Integration von Amazon Inspector mit Microsoft Azure

Sie können in Amazon Inspector eine Integration zu Microsoft Azure erstellen, um Softwareschwachstellen in Ihren Azure-Ressourcen zu erkennen. Nachdem Sie den Connector erstellt haben, scannt Amazon Inspector kontinuierlich Ihre Azure Virtual Machines (VMs), Azure Function Apps und Azure Container Registry (ACR) -Images auf bekannte Sicherheitslücken und generiert Ergebnisse mit Anleitungen zur Behebung.

Funktionsweise

Eine Integration in Microsoft Azure ermöglicht es Amazon Inspector, Ihre Azure-Ressourcen zu entdecken und nach Softwareschwachstellen zu durchsuchen. Der Connector verwendet IAM Outbound Identity Federation, um sich sicher in Ihrer Azure-Umgebung zu authentifizieren, ohne langlebige Anmeldeinformationen zu speichern.

1. Sie erstellen einen Connector in der Amazon Inspector Inspector-Konsole oder mithilfe der API. Sie geben Ihre Azure-Mandanten-ID, Azure-Abonnements und Azure-Regionen an und wählen aus, welche Funktionen aktiviert werden sollen (VM-Scan, Funktions-App-Scanning, ACR-Scan).
2. Sie führen das generierte Azure-Setupskript aus, um die Azure-App-Registrierung, die Event Hub-Infrastruktur und die Berechtigungen zu konfigurieren.
3. Amazon Inspector erkennt Ihre Azure-Ressourcen und beginnt mit der Suche nach Sicherheitslücken.

Wie AWS Config wird verwendet

AWS Config wird intern verwendet, um Daten zur Ressourcenkonfiguration zu sammeln und Änderungen in Ihrer Azure-Umgebung zu erkennen (z. B. das Starten neuer VMs oder das Übertragen neuer Images auf ACR). Der Connector verarbeitet dies automatisch. Sie müssen den Recorder nicht AWS Config separat aktivieren, konfigurieren oder für die AWS Config Nutzung bezahlen. Diese Kosten sind in Ihren Amazon Inspector Inspector-Preisen für Azure-Ressourcen enthalten.

Arten von Anschlüssen

Amazon Inspector unterstützt zwei Konnektortypen für Integrationen mit Microsoft Azure:

- Service-linked Connector (SLC) — Wird automatisch erstellt, wenn Sie eine Integration mit Microsoft Azure einrichten. AWS Security Hub CSPM Sie können einen SLC nicht direkt in Amazon Inspector bearbeiten. Verwalte es AWS Security Hub CSPM stattdessen über. Der AWS Security Hub CSPM Geltungsbereich (die überwachten Azure-Abonnements und überwachten Azure-Regionen) muss immer größer oder gleich dem Umfang aller vorhandenen, vom Kunden verwalteten Connectoren sein.
- Customer-managed Connector (CMC) — Direkt in der Amazon Inspector Inspector-Konsole oder über API erstellt. Sie haben die volle Kontrolle über Umfang und Funktionen, einschließlich der Möglichkeit, den Connector zu erstellen, zu ändern und zu löschen.

Für die eigenständige Nutzung von Amazon Inspector erstellen Sie einen vom Kunden verwalteten Connector. Wenn Sie bereits über eine AWS Security Hub CSPM Integration mit Microsoft Azure verfügen, erhält Amazon Inspector automatisch einen serviceverknüpften Connector, sodass Sie keinen separaten erstellen müssen.

Methoden zum Scannen

Amazon Inspector verwendet die folgenden Methoden, um Azure-Ressourcen zu scannen:

- Virtuelle Maschinen — Amazon Inspector verwendet den Amazon Inspector VM Scanner-Agenten, um VMs auf Betriebssystem- und Anwendungsschwachstellen zu scannen. Sie können zwischen verwalteter Bereitstellung (empfohlen) und manueller Installation wählen. Weitere Informationen zum Agenten finden Sie unterAmazon Inspector VM-Scanner.
- Funktions-Apps — Amazon Inspector extrahiert und analysiert automatisch Funktionscodeabhängigkeiten auf bekannte Sicherheitslücken.
- Container Registry-Images — Amazon Inspector scannt Container-Images, die in Azure Container Registry (ACR) gespeichert sind, automatisch auf Sicherheitslücken in Betriebssystempaketen und Anwendungsabhängigkeiten.

Voraussetzungen für die Integration von Amazon Inspector mit Microsoft Azure

Bevor Sie einen Connector erstellen, stellen Sie sicher, dass Ihr AWS Konto und Ihre Azure-Umgebung die folgenden Anforderungen erfüllen. Das Azure-Setupskript verarbeitet die meisten

Azure-Konfigurationen automatisch (App-Registrierung, Event Hub, Diagnoseeinstellungen, Azure-Rollenzuweisungen und Microsoft Graph API-Berechtigungen).

AWS Kontoeinrichtung — alle Funktionen

IAM Outbound Identity Federation und OIDC-Anbieter

Für Ihr AWS Konto muss eine IAM-Token-Aussteller-URL für den ausgehenden Identitätsverbund konfiguriert sein. Auf diese Weise kann Amazon Inspector sich mithilfe kurzlebiger Token sicher in Ihrer Azure-Umgebung authentifizieren.

Für das Scannen virtueller Maschinen benötigen Sie außerdem einen IAM-OIDC-Identitätsanbieter, der mit der Aussteller-URL konfiguriert ist. `sts.windows.net/`*`tenant-id`*`/` Dieser Anbieter ermöglicht es dem Amazon Inspector VM Scanner-Agent auf Ihren Azure-VMs, sich wieder bei der Verwendung zu authentifizieren. AWS `sts:AssumeRoleWithWebIdentity` Ohne diesen Anbieter kann der VM-Scanner keine Telemetrie an Amazon Inspector senden.

Wenn Sie den ausgehenden Identitätsverbund bereits für einen anderen AWS Sicherheitsdienst (z. B. AWS Security Hub CSPM oder AWS Security Hub CSPM CSPM) konfiguriert haben, müssen Sie ihn nicht erneut konfigurieren. Weitere Informationen finden Sie unter Identitätsanbieter und Verbund im IAM-Benutzerhandbuch.

Auf Azure-Seite erstellt das Setup-Skript die föderierten Identitätsdaten, die dem STS-OIDC-Aussteller vertrauen. AWS Sie müssen dies nicht manuell konfigurieren.

Service-linked Rolle

Die API erstellt die mit dem Service verknüpfte Amazon Inspector-Rolle im Namen des Kunden bei der Connector-Erstellung. Eine manuelle Bereitstellung von Spiegelreflexkameras ist nicht erforderlich.

Konnektor konfigurieren

Ein Connector-ARN ist erforderlich, wenn ein Connector über die API erstellt wird. Der Config-Connector erfasst Änderungen an der Azure-Ressourcenkonfiguration. In der Konsole können Sie entweder einen vorhandenen Connector auswählen oder während der Connector-Erstellung einen neuen erstellen.

IAM-Berechtigungen für die Konsole

Der IAM-Prinzipal (Benutzer oder Rolle), der den Connector erstellt, benötigt die folgenden Berechtigungen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "config:GetConnector",
        "config:ListConnectors",
        "config:PutConnector",
        "inspector2:CreateConnector",
        "inspector2:DeleteConnector",
        "inspector2:ListConnectors",
        "inspector2:ListTagsForResource",
        "inspector2:TagResource",
        "inspector2:UntagResource",
        "inspector2:UpdateConnector"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS Kontoeinrichtung — nur VM-Scannen

Zusätzlich zu den oben genannten Anforderungen erfordert das Scannen virtueller Maschinen:

- Amazon EC2 Systems Manager ist in derselben Region wie der Connector konfiguriert
- Amazon EC2 Systems Manager Automation-Berechtigungen für die verwaltete Bereitstellung des Amazon Inspector VM Scanner-Agenten

Einrichtung eines Azure-Kontos

Bevor Sie das Azure-Setupskript ausführen, stellen Sie sicher, dass Sie über Folgendes verfügen:

Azure-Mandant

Ein kommerzieller Microsoft Azure-Mandant. Azure Government und Azure China werden nicht unterstützt.

Microsoft Entra IDVerfügbarkeit protokollieren

Für die Ressourcenaufzeichnung sind Microsoft Graph Prüfprotokolle und Anmeldeprotokolle von Ihrem Microsoft Entra ID Mandanten erforderlich. Für einige Microsoft Entra ID Protokolle ist eine Microsoft Entra ID P1- oder P2-Lizenz erforderlich.

Wenn dem Mandanten die erforderlichen Protokolle fehlen, werden bei der Ressourcenaufzeichnung möglicherweise Teilfehler gemeldet. Die App-Registrierung und die Azure-Rollenzuweisungen liefern keine fehlenden Protokolldaten.

Azure-Rolle

Die Person, die Azure konfiguriert, muss die globale Administratorrolle im Azure-Mandanten haben.

Azure-CLI

Die Azure-CLI wurde installiert und authentifiziert (`az login`).

> **Note**
>
> Die Person, die die Azure-Umgebung einrichtet, muss nicht dieselbe Person sein, die den Connector in der Amazon Inspector-Konsole erstellt hat.

Konfiguration von Microsoft Azure für die Integration mit Amazon Inspector

Der Azure-Einrichtungsprozess konfiguriert Ihre Azure-Umgebung so, dass Amazon Inspector Benachrichtigungen über Ressourcenänderungen authentifizieren und empfangen kann. Amazon Inspector generiert während der Connector-Erstellung ein benutzerdefiniertes Setup-Skript. Das Skript führt die folgenden Schritte aus.

> **Note**
>
> Wenn Sie die Einrichtung der Azure-Umgebung für AWS Security Hub CSPM oder AWS Security Hub CSPM CSPM bereits abgeschlossen haben, wird dieselbe App-Registrierung gemeinsam genutzt. Das Skript erkennt die bestehende Registrierung und fügt die

erforderlichen Federated Identity-Anmeldeinformationen für Amazon Inspector hinzu, ohne die bestehende Konfiguration zu ändern.

Was macht das Setup-Skript

Das Setup-Skript führt eine Reihe von Aufgaben aus. In den folgenden Abschnitten werden alle Aufgaben beschrieben, die das Skript ausführt. Wenn Ihre Organisation die Ausführung von Skripten einschränkt, können Sie diese Aufgaben manuell ausführen und dabei dieses Thema als Leitfaden verwenden.

Schritt 1: Registrieren Sie eine Azure-Anwendung

Registrieren Sie eine neue Anwendung inMicrosoft Entra ID. Diese Anwendung dient als Identität, AWS die für die Authentifizierung in Ihrer Azure-Umgebung mithilfe von Verbundanmeldedaten verwendet wird. Es sind keine Client-Geheimnisse erforderlich.

```
$ az ad app create --display-name "AWSSecurityHubIntegration"
```

Notieren Sie sich die Anwendungs-ID (Client) aus der Ausgabe (das `appId` Attribut). Sie geben diesen Wert an, wenn Sie den Connector in der Amazon Inspector Inspector-Konsole erstellen.

Schritt 2: Erstellen Sie einen Service Principal

Erstellen Sie einen Dienstprinzipal für die registrierte Anwendung. Sie verwenden Azure-Rollenzuweisungen, um dem Dienstprinzipal Berechtigungen zu erteilen. Der Dienstprinzipal führt Aktionen in Ihrer Azure-Umgebung durch.

```
$ az ad sp create --id application-client-id
```

Wo *`application-client-id`* ist die Anwendungs-ID (Client) aus dem vorherigen Schritt.

Schritt 3: Konfigurieren Sie Anmeldeinformationen für föderierte Identitäten

Konfigurieren Sie föderierte Identitätsanmeldedaten in der Anwendung, um eine Authentifizierung ohne geheime Clientdaten AWS zu ermöglichen. Dadurch wird eine OIDC-Vertrauensstellung zwischen und Azure eingerichtet. AWS Es werden keine langlebigen Geheimnisse oder Anmeldeinformationen gespeichert.

```
$ az ad app federated-credential create \
  --id application-client-id \
  --parameters '{
    "name": "AWSConfigFederation",
    "issuer": "token-issuer-url",
    "subject": "arn:aws:iam::account-id:role/aws-service-role/
thirdparty.config.amazonaws.com/AWSServiceRoleForConfigThirdParty",
    "audiences": ["api://AzureADTokenExchange"],
    "description": "Federation for AWS Config third-party cloud resource discovery"
  }'
```

Die wichtigsten Werte:

- Aussteller — Ihre Token-Aussteller-URL von IAM Outbound Identity Federation. Dies folgt dem Format. `https://uuid.tokens.sts.global.api.aws`
- Betreff — Der ARN der `AWSServiceRoleForConfigThirdParty` serviceverknüpften Rolle in Ihrem AWS-Konto. AWS erstellt diese dienstverknüpfte Rolle automatisch, wenn Sie den Connector erstellen.
- Zielgruppe — `api://AzureADTokenExchange` (Standardwert für den Workload-Identitätsverbund).

Schritt 4: Weisen Sie die Rolle Reader zu

Weisen Sie dem Dienstprinzipal im Bereich der Mandantenstammverwaltungsgruppe die Rolle Reader zu. Dadurch wird schreibgeschützter Zugriff auf alle Azure-Ressourcen für alle Abonnements im Mandanten gewährt.

```
$ az role assignment create \
  --assignee application-client-id \
  --role "Reader" \
  --scope "/providers/Microsoft.Management/managementGroups/tenant-id"
```

> **Note**
>
> Wenn dieser Schritt fehlschlägt, stellen Sie sicher, dass die Zugriffsverwaltung für Azure-Ressourcen unter Microsoft Entra ID > Eigenschaften auf Ja gesetzt ist. Melden Sie sich dann ab und wieder an, um Ihr Token zu aktualisieren.

Schritt 5: Microsoft Graph API-Berechtigungen konfigurieren

Erteilen Sie der Anwendung die erforderlichen Microsoft Graph API-Berechtigungen (Anwendungstyp): `Directory.Read.All`, `AuditLog.Read.All` und `Policy.Read.All`. Erteilen Sie dann die Zustimmung des Administrators.

Schritt 6: Azure Event Hub einrichten

AWS Config verwendet einen Azure Event Hub, um nahezu in Echtzeit Benachrichtigungen über Änderungen der Ressourcenkonfiguration in Ihrer Azure-Umgebung zu erhalten. Das Setup-Skript erstellt die Event Hub-Infrastruktur und markiert sie für die Erkennung.

```
$ az group create \
  --name "aws-securityhub-integration" \
  --location event-hub-region

$ az eventhubs namespace create \
  --resource-group "aws-securityhub-integration" \
  --name "aws-securityhub-account-id" \
  --location event-hub-region \
  --sku Standard

$ az eventhubs eventhub create \
  --resource-group "aws-securityhub-integration" \
  --namespace-name "aws-securityhub-account-id" \
  --name "activitylog" \
  --message-retention 1 \
  --partition-count 4

$ az eventhubs eventhub consumer-group create \
  --resource-group "aws-securityhub-integration" \
  --namespace-name "aws-securityhub-account-id" \
  --eventhub-name "activitylog" \
  --name "AWSConfig"
```

Nachdem Sie den Event Hub erstellt haben, kennzeichnen Sie den Namespace für die Erkennung und weisen Sie ihm die Datenempfängerrolle zu:

```
$ az tag create \
  --resource-id namespace-resource-id \
  --tags "AWSConfig-account-id-region=activitylog"
```

```
$ az role assignment create \
  --assignee application-client-id \
  --role "Azure Event Hubs Data Receiver" \
  --scope namespace-resource-id
```

Schlüsselkonfiguration:

- Namespace-SKU — Standard (für Verbrauchergruppen erforderlich).
- Name des Event Hubs —`activitylog`.
- Verbrauchergruppe —`AWSConfig`.
- Tag — `AWSConfig-`*`account-id`*`-`*`region`* mit Wert`activitylog`. Dieses Tag ermöglicht es AWS Config , den Event Hub automatisch zu entdecken.
- Rollenzuweisung — Azure Event Hubs Data Receiver im Namespace.

> **Note**
>
> Das Tag-Format steht `AWSConfig-`*`account-id`*`-`*`region`* für Ihre 12-stellige AWS Konto-ID und *`region`* für die AWS Region, in der Sie den Connector erstellt haben (z. B.). *`account-id`* `us-east-1`

Schritt 7: Konfigurieren Sie den Export des Aktivitätsprotokolls und des Entra ID Auditprotokolls

Konfigurieren Sie Azure für den Export von Aktivitätsprotokollen und Entra ID Auditprotokollen in den Event Hub.

Export des Aktivitätsprotokolls:

Konfigurieren Sie die Diagnoseeinstellungen für jedes Abonnement, um Aktivitätsprotokolle in den Event Hub zu exportieren. Das Setup-Skript konfiguriert dies automatisch für alle überwachten Abonnements. Aktivitätsprotokolle erfassen Aktivitäten auf Abonnementebene, einschließlich Rollenzuweisungen und Ressourcenänderungen.

Entra IDExport des Auditprotokolls:

Konfigurieren Sie die DiagnoseeinstellungenMicrosoft Entra ID, um Auditprotokolle in denselben Event Hub zu exportieren. Navigieren Sie zu Microsoft Entra ID> Überwachung >

Diagnoseeinstellungen > Diagnoseeinstellung hinzufügen. Aktivieren Sie sie AuditLogsund leiten Sie sie an Ihren Event Hub weiter.

Das Setup-Skript ausführen

So führen Sie das Azure-Setup-Skript aus:

1. Öffnen Sie ein Terminal mit der Azure-CLI, die für Ihren Azure-Mandanten authentifiziert ist (`az login`). Sie müssen über Mitwirkende- und Eigentümerberechtigungen für die Zielabonnements verfügen.
2. Kopieren Sie das Setup-Skript aus der Amazon Inspector Inspector-Konsole (generiert bei der Connector-Erstellung).
3. Führen Sie das Setup-Skript aus. Das Skript vervollständigt die oben beschriebene Einrichtung der Azure-Umgebung.
4. Nachdem das Skript erfolgreich abgeschlossen wurde, kehren Sie zur Amazon Inspector Inspector-Konsole zurück, um die Erstellung des Connectors abzuschließen.

> **Important**
>
> Wenn Sie später den Konnektorbereich ändern (z. B. eine neue Azure-Region hinzufügen), generiert die Konsole das Setup-Skript neu, um die Event Hub-Infrastruktur in der neuen Region bereitzustellen. Sie müssen das aktualisierte Skript erneut ausführen.

Konfiguration von Amazon Inspector für die Integration mit Microsoft Azure

Nachdem Sie Ihre Azure-Umgebung konfiguriert haben, erstellen Sie einen Connector in Amazon Inspector, um mit dem Scannen Ihrer Azure-Ressourcen auf Sicherheitslücken zu beginnen.

Um einen Azure-Connector zu erstellen

1. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Integrationen aus.
3. Wählen Sie Konnektor erstellen.

4. Geben Sie die folgenden Informationen ein:

   - Name — Ein eindeutiger Name für diesen Connector. Längenbeschränkungen für Namen und Beschreibung finden Sie unter dem CreateConnector Vorgang in der Amazon Inspector API-Referenz.
   - Beschreibung (optional) — Eine Beschreibung dieses Connectors.

   > **Important**
   >
   > Namen und Beschreibungen werden verwendet, um Ihre Inhalte zu identifizieren. Wir empfehlen Ihnen, keine sensiblen, vertraulichen oder persönlich identifizierbaren Informationen (PII) darin zu verwenden.

   - Azure Tenant ID — Ihre Microsoft Entra ID Mandanten-ID. Zu finden im Azure-Portal Microsoft Entra ID unter Eigenschaften.
5. Wählen Sie die zu aktivierenden Funktionen aus:

   - VM-Scannen (Amazon Inspector VM Scanner-Agent, der auf Azure-VMs bereitgestellt wird)
   - Funktion App-Scanning (Analyse der Codeabhängigkeit)
   - Scannen von ACR-Bildern (Erkennung von Sicherheitslücken in Container-Images)

   Sie können eine beliebige Kombination aktivieren. Funktionen können später über geändert werden`UpdateConnector`.
6. Konfigurieren Sie Ihre Überwachungsziele:

   - Azure-Abonnements — Wählen Sie aus, ob Sie alle Abonnements in Ihrem Mandanten überwachen oder einzelne Abonnement-IDs angeben möchten.
   - Azure-Regionen — Wählen Sie aus, ob Sie alle Azure-Regionen oder bestimmte Regionen aktivieren möchten.
7. Konfigurieren Sie die Aufzeichnungsquelle:

   Wählen Sie eine vorhandene Aufnahmequelle aus oder erstellen Sie eine neue. Der ARN der Aufzeichnungsquelle ist ein Pflichtfeld in der API (`inspector2:CreateConnector`).
8. Kopieren Sie das generierte Azure-Setupskript, führen Sie es in Ihrer Azure-Umgebung aus (siehethe section called “Azure konfigurieren”) und wählen Sie dann Connector erstellen aus.

Verwaltete Bereitstellung des Amazon Inspector VM Scanners

Wenn Sie das Scannen von VMs aktiviert haben, verwendet Amazon Inspector Amazon EC2 Systems Manager Automation, um den Amazon Inspector VM Scanner-Agenten auf Ihren Azure-VMs bereitzustellen und zu verwalten. Bei verwalteter Bereitstellung:

- Es ist keine manuelle Installation erforderlich.
- Der Agent bleibt automatisch auf dem neuesten Stand.
- Amazon Inspector verwaltet den gesamten Lebenszyklus (Installation, Aktualisierung, Zustandsüberwachung).

Amazon Inspector verwendet Amazon EC2 Systems Manager Automation, um den Agenten bereitzustellen und zu verwalten.

Um das Scannen von VMs zu aktivieren, müssen Sie die folgenden IAM-Ressourcen in Ihrem Konto erstellen.

IAM OIDC-Identitätsanbieter

Erstellen Sie einen IAM-OIDC-Identitätsanbieter für mit der folgenden Konfiguration: Microsoft Entra ID

- URL des Ausstellers: `https://sts.windows.net/`*`tenant-id`*`/`
- Kunden-ID-Liste:`api://`*`app-id`*, `api://AzureADTokenExchange`
- Fingerabdruck: Eine 40-stellige Zeichenfolge, die ausschließlich Nullen enthält (40 Mal `0` wiederholt). AWS STS validiert Token über den JWKS-Endpunkt und nicht über den Fingerabdruckwert.

Inspector2VmScannerRole-***`tenant-id`***

Wird vom Amazon Inspector VM Scanner-Agent verwendet, der auf Azure-VMs ausgeführt wird, um Scan-Telemetrie an Amazon Inspector zu senden. Diese Rolle verwendet den OIDC-Web-Identitätsverbund über den Identitätsanbieter. Microsoft Entra ID

Vertrauensrichtlinie:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::account-id:oidc-provider/
sts.windows.net/tenant-id/"
            },
            "Action": "sts:AssumeRoleWithWebIdentity"
        }
    ]
}
```

Genehmigungsrichtlinie (TelemetryPolicy):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "inspector2-telemetry:*",
            "Resource": "*"
        }
    ]
}
```

Schlagworte:

- `inspector-managed` = `true`

Inspector2SSMFederationRole-*tenant-id*

Aktiviert den OIDC-Verbund mit Azure für eine sichere cloudübergreifende Authentifizierung. Diese Rolle wird von AssumeRole und und DispatchRole von Amazon EC2 Systems Manager übernommen, um Web-Identitätstoken für Azure-Operationen abzurufen.

Vertrauensrichtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::account-id:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalTag/caller": "SSM"
                },
                "ArnLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::account-id:role/Inspector2SSMAssumeRole-tenant-
id",
                        "arn:aws:iam::account-id:role/Inspector2SSMDispatchRole-tenant-
id"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Genehmigungsrichtlinie (FederationPolicy):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "sts:IdentityTokenAudience": [
                        "api://AzureADTokenExchange"
                    ]
                }
            },
            "Action": [
                "sts:GetWebIdentityToken",
```

```
                "sts:TagGetWebIdentityToken"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Schlagworte:

- `caller` = `SSM`
- `inspector-managed` = `true`

Inspector2SSMAssumeRole-*tenant-id*

Führt Amazon EC2 Systems Manager Automation aus, um Azure-VMs als verwaltete Instanzen zu integrieren. Diese Rolle erstellt und verwaltet Hybrid-Aktivierungen, übergibt Rollen an Amazon EC2 Systems Manager und ruft Automatisierungsdokumente ab.

Vertrauensrichtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                }
            }
        }
    ]
}
```

Genehmigungsrichtlinie (AssumeRolePolicy):

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::account-id:role/
Inspector2SSMFederationRole-tenant-id"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/
AmazonEC2RunCommandRoleForManagedInstances",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateActivation",
                "ssm:DeleteActivation",
                "ssm:DescribeActivations",
                "ssm:DescribeInstanceInformation",
                "ssm:PutConfigurePackageResult",
                "iam:ListRoleTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:AddTagsToResource",
            "Resource": [
                "arn:aws:ssm:*:account-id:activation/*",
                "arn:aws:ssm:*:account-id:managed-instance/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssm:GetDocument",
            "Resource": [
                "arn:aws:ssm:*:*:document/AWS-InstallDistributorPackageOnAzure",
```

```
                "arn:aws:ssm:*:*:document/AWS-InstallSsmAgentOnAzure",
                "arn:aws:ssm:*::document/AmazonInspector2-InspectorVmScanner"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssm:GetCloudConnector",
            "Resource": "*"
        }
    ]
}
```

Schlagworte:

- `caller` = `SSM`
- `AzureTenantId` = *`tenant-id`*
- `AzureApplicationId` = *`app-id`*
- `inspector-managed` = `true`

Inspector2SSMDispatchRole-***`tenant-id`***

Löst Azure-VM-Ziele auf und leitet die Installation des Amazon Inspector VM Scanner-Agenten über Amazon EC2 Systems Manager Automation ein. Diese Rolle startet Automatisierungsausführungen und leitet AssumeRole sie an Amazon EC2 Systems Manager weiter.

Vertrauensrichtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                }
            }
```

```
        }
    ]
}
```

Genehmigungsrichtlinie (DispatchPolicy):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/Inspector2SSMAssumeRole-tenant-
id",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": [
                "arn:aws:ssm:*:*:document/AWS-InstallDistributorPackageOnAzure",
                "arn:aws:ssm:*:account-id:automation-execution/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::account-id:role/
Inspector2SSMFederationRole-tenant-id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:ListCloudConnectors",
                "ssm:GetCloudConnector",
                "iam:ListRoleTags"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Schlagworte:

- `caller` = `SSM`
- `inspector-managed` = `true`

Bei der Azure-App-Registrierung muss der Anwendungs-ID-URI auf gesetzt sein`api://app-id`.

Für jede Azure-Ziel-VM muss eine vom System zugewiesene verwaltete Identität aktiviert sein. Verwenden Sie eine DeployIfNotExists Richtlinie in Ihrer Azure-Umgebung, um neuen VMs automatisch verwaltete Identitäten zuzuweisen.

Weitere Informationen zum Amazon Inspector VM Scanner-Agenten finden Sie unterAmazon Inspector VM-Scanner.

Alternativ können Sie den Amazon Inspector VM Scanner-Agenten manuell installieren. Verwenden Sie für eine manuelle Installation in großem Umfang auf vielen Azure-VMs eine benutzerdefinierte Azure-VM-Erweiterung. Detaillierte Anweisungen finden Sie unter Amazon Inspector VM-Scanner.

Um bestimmte Azure-VMs vom Scannen auszuschließen, wenden Sie das folgende Tag auf die VM an:

- Schlüssel: `InspectorExclusion`
- Value (Wert): `true`

Den Geltungsbereich eines Connectors anpassen

Nachdem Sie einen Connector erstellt haben, können Sie seinen Umfang anpassen, indem Sie die Azure-Abonnements oder Azure-Regionen ändern, die er überwacht. Wenn Sie den Umfang anpassen, werden die meisten Änderungen innerhalb von etwa 15 Minuten wirksam.

Bevor Sie den Umfang anpassen, sollten Sie Folgendes beachten:

- Wenn Sie dem Bereich eine Region hinzufügen, generiert die Konsole ein aktualisiertes Setup-Skript, das die erforderliche Event Hub-Infrastruktur in der neuen Region erstellt.

- Wenn ein serviceverknüpfter Connector existiert (erstellt über AWS Security Hub CSPM), können Sie seinen Umfang nicht direkt in Amazon Inspector reduzieren. Ändern Sie AWS Security Hub CSPM stattdessen den Bereich in.
- Der Umfang eines AWS Security Hub CSPM Connectors darf niemals geringer sein als der eines vorhandenen, vom Kunden verwalteten Connectors in Amazon Inspector für denselben Azure-Mandanten.

Der Konnektor wird verifiziert

Nachdem Sie den Connector erstellt haben, beginnt Amazon Inspector mit der Erkennung und dem Scannen Ihrer Azure-Ressourcen. Es kann bis zu 24 Stunden dauern, bis die ersten Ergebnisse angezeigt werden. Gehen Sie wie folgt vor, um zu überprüfen, ob der Connector funktioniert:

1. Navigieren Sie in der Amazon Inspector Inspector-Konsole zu Zusammenfassung, um zu überprüfen, ob die Azure-Ressourcenzahlen aufgefüllt werden.
2. Navigieren Sie zu Coverage, um zu überprüfen, ob Azure-Ressourcen mit dem Scan-Status angezeigt werden.
3. Navigieren Sie zu Integrationen, wählen Sie Ihre Integration in Microsoft Azure aus und überprüfen Sie, ob der Connector-Status angezeigt wird`Active`.
4. Stellen Sie für das Scannen virtueller Computer sicher, dass Azure-VMs in der Ansicht „Abdeckung“ mit dem Agentenstatus Amazon Inspector VM Scanner angezeigt werden.
5. Navigieren Sie zu Ergebnisse und filtern Sie nach Ressourcentyp, um die Ergebnisse der Azure-Ressourcen anzuzeigen.

> **Note**
>
> Der Integritätsstatus des Connectors ist letztendlich konsistent. Wenn ein Berechtigungs- oder Konfigurationsproblem auftritt, ändert sich der Status des Connectors schnell in Heruntergestuft und es wird eine umsetzbare Meldung angezeigt. Nachdem Sie das Problem behoben haben, kann es jedoch bis zu 24 Stunden dauern, bis der Status wieder zu Verbunden zurückkehrt.
>
> - Ein Status „Heruntergestuft“ nach einem kürzlich erfolgten Update weist nicht unbedingt auf ein anhaltendes Problem hin.

- Überprüfen Sie die CloudWatch Metriken im Namespace, um in Echtzeit Einblick in die Fehler bei der Aufzeichnung zu erhalten.

Fehlerbehebung

Wenn der Connector-Status Fehler anzeigt oder Ressourcen nach 24 Stunden nicht angezeigt werden, überprüfen Sie Folgendes:

Der Connector zeigt den Status Fehler an

Stellen Sie sicher, dass die Azure-App-Registrierung über die richtigen Anmeldeinformationen für die Verbundidentität verfügt und dass die URL des IAM-Token-Ausstellers in Ihrem Konto konfiguriert ist. AWS

Einige Azure-Ressourcen konnten nicht aufgezeichnet werden

Der Fehler kann aus einem der folgenden Gründe auftreten:

- Die Azure-Umgebung erfüllt nicht die Voraussetzungen für die Ressourcenaufzeichnung.
- Für die App-Registrierung ist keine Administrator-Zustimmung für Microsoft Graph API-Berechtigungen erforderlich.
- Die Rollenzuweisung „Leser“ deckt nicht alle erforderlichen Bereiche ab.
- Der Microsoft Entra ID Mandant verfügt nicht über die erforderlichen Protokolldaten.

Gehen Sie wie folgt vor, um den Fehler zu beheben:

1. Um zu überprüfen, ob Ihre Umgebung alle Anforderungen erfüllt, lesen Sie die Voraussetzungen für die Integration von Amazon Inspector mit Microsoft Azure.
2. Vergewissern Sie sich mit Zustimmung des Administrators, dass die App-Registrierung über die erforderlichen Microsoft Graph API-Berechtigungen verfügt.
3. Stellen Sie sicher, dass der Dienstprinzipal die Rolle Reader im Bereich der Stammverwaltungsgruppe des Mandanten innehat.

Ressourcen werden nicht angezeigt

Vergewissern Sie sich, dass der Event Hub Ereignisse empfängt. Überprüfen Sie die Event Hub-Metriken im Azure-Portal und stellen Sie sicher, `AWSConfig` dass die Nutzergruppe existiert.

VMs werden nicht gescannt

Stellen Sie sicher, dass virtuelle Maschinen über Netzwerkkonnektivität zu Amazon EC2 Systems Manager Manager-Endpunkten verfügen und ein unterstütztes Betriebssystem ausführen.

ACR-Bilder werden nicht angezeigt

Stellen Sie sicher, dass die Diagnoseeinstellungen in Ihren Container-Registern aktiviert sind.

Die Änderung des Umfangs wird nicht wirksam

Wenn Sie den Konnektorbereich ändern, um eine neue Azure-Region hinzuzufügen, generiert die Konsole das Setup-Skript neu. Sie müssen das aktualisierte Skript erneut ausführen, um die Event Hub-Infrastruktur in der neuen Region bereitzustellen.

Wie Amazon Inspector mit Ressourcen-Identifikatoren umgeht

Wenn Sie Ihre Azure-Integration für Amazon Inspector aktivieren, werden Ressourcen-IDs von anderen Cloud-Anbietern in AWS Config Amazon Inspector und anderen AWS Diensten (je nach Bedarf) als Metadaten gespeichert, die sich auf die Verwaltung der entsprechenden Ressourcenkonfigurationsdaten beziehen, die von den anderen Cloud-Anbietern gesammelt wurden. Solche Ressourcen-Identifikatoren stellen nicht Ihre Inhalte dar, und wir empfehlen Ihnen, keine sensiblen, vertraulichen oder persönlich identifizierbaren Informationen in sie aufzunehmen.

Die folgenden Identifikatoren aus Ihrer verbundenen Cloud-Umgebung werden gespeichert und verwendet, AWS um Multi-Cloud-Sicherheitsfunktionen bereitzustellen:

- Ressourcen-IDs: Azure-Mandanten-ID, Abonnement-ID, Standort (Region), Ressourcen-ID (Ressourcengruppen-IDs oder -namen, Ressourcenanbieter, Ressourcentyp)

Beziehungen zwischen diesen Identifikatoren — einschließlich der Beziehung zwischen Ressourcen und der Beziehung der Ergebnisse zu Ressourcen — werden ebenfalls als Dienstmetadaten gespeichert. AWS verwendet diese Identifikatoren für die Korrelation von Ressourcen, für die Zuordnung von Ergebnissen zu Ressourcen, für die Protokollierung von Dienstabläufen und für die Deduplizierung.

Löschen eines Azure-Connectors in Amazon Inspector

Sie können einen Azure-Connector löschen, wenn Sie nicht mehr möchten, dass Amazon Inspector Ressourcen in diesem Azure-Mandanten scannt.

Was passiert, wenn Sie einen Connector löschen

Wenn Sie einen Azure-Connector löschen:

- Amazon Inspector beendet das Scannen aller mit dem Connector verknüpften Azure-Ressourcen.
- Bestehende Ergebnisse für Azure-Ressourcen wurden geschlossen.
- Azure-Ressourcen werden nicht mehr in der Coverage-Ansicht angezeigt.
- Wenn ein dienstgebundener Connector vorhanden ist (erstellt über AWS Security Hub CSPM), hat das Löschen eines vom Kunden verwalteten Connectors keine Auswirkungen auf den dienstverknüpften Connector. Amazon Inspector scannt weiter über den serviceverknüpften Connector.
- Die Azure-App-Registrierung, die Event Hub-Infrastruktur und die in Ihrer Azure-Umgebung konfigurierten Berechtigungen werden nicht automatisch entfernt. Sie müssen Azure-Ressourcen manuell bereinigen, wenn sie von anderen AWS Diensten nicht mehr benötigt werden.

> **Note**
>
> Wenn Sie die Azure-App-Registrierung mit AWS Security Hub CSPM oder AWS Security Hub CSPM CSPM teilen, löschen Sie die App-Registrierung nicht, nachdem Sie den Amazon Inspector-Connector entfernt haben. Andere Dienste hängen immer noch davon ab.

Löschen eines Connectors

Um einen Azure-Connector zu löschen

1. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Integrationen aus.
3. Wählen Sie den Azure-Connector aus, den Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Geben Sie im Bestätigungsdialogfeld Connector löschen ein **`delete`** und wählen Sie ihn aus.

Sie können einen Connector auch mithilfe der `inspector2:DeleteConnector` API-Operation löschen.

Important

Wenn Sie einen Connector löschen und innerhalb von 6 Stunden einen neuen erstellen, kann die Ressourcenerkennung bis zu 6 Stunden dauern. Planen Sie das Löschen und Wiederherstellen eines Connectors entsprechend ein.

Amazon Inspector SBOM-Generator

Eine Softwareliste (SBOM) ist eine formal strukturierte Liste von Komponenten, Bibliotheken und Modulen, die zur Erstellung einer Software erforderlich sind. Der Amazon Inspector SBOM Generator (Sbomgen) ist ein Tool, das eine SBOM für Archive, Container-Images, Verzeichnisse, lokale Systeme sowie kompilierte Dateien und Binärdateien erstellt. Go Rust Sbomgensucht nach Dateien, die Informationen über installierte Pakete enthalten. Wenn Sbomgen eine relevante Datei gefunden wird, extrahiert es Paketnamen, Versionen und andere Metadaten. Sbomgenwandelt dann Paketmetadaten in eine CycloneDX SBOM um. Sie können Sbomgen damit die CycloneDX SBOM als Datei oder in STDOUT generieren und SBOMs zur Schwachstellenerkennung an Amazon Inspector senden. Sie können es auch Sbomgen als Teil der CI/CD Integration verwenden, bei der Container-Images automatisch als Teil Ihrer Bereitstellungspipeline gescannt werden.

Amazon Inspector Sbomgen verwendet mehrere verwaltete Scanfunktionen. Wenn Amazon Inspector Amazon ECR-Container-Images, AWS Lambda-Funktionsscans und agentenloses Scannen von Amazon Elastic Compute Cloud-Instances mithilfe der Amazon EBS-Volumenanalyse durchführt, wird Sbomgen hinter den Kulissen das Softwareinventar generiert, das für die Schwachstellenbewertung verwendet wird. Das auf dieser Seite beschriebene eigenständige Sbomgen Tool bietet dieselbe grundlegende Technologie, sodass Sie die SBOM-Generierung direkt in Ihre Workflows integrieren können.

Unterstützte Pakettypen

Sbomgensammelt Inventar für die folgenden Pakettypen:

- Alpine APK
- Debian/Ubuntu DPKG
- Red Hat RPM
- Homebrew
- C#
- Go
- Java
- Node.js
- PHP
- Python

- Ruby
- Rust

Unterstützte Konfigurationsprüfungen für Container-Images

Sbomgenkann eigenständige Docker-Dateien scannen und den Verlauf anhand vorhandener Images auf Sicherheitsprobleme aufbauen. Weitere Informationen finden Sie unter Amazon Inspector Dockerfile-Checks.

Sbomgenkann mit dem Argument auch unterstützte CIS Docker Benchmark Prüfungen für ein Container-Image auswerten und anhand des `--enable-cis-docker` Arguments Indikatoren aufzeichnen, die die FIPS-related kryptografische Konfiguration eines Systems beschreiben, bei Linux Container- und Localhost-Scans. `--enable-fips-checks` Beide Argumente sind experimentell und standardmäßig deaktiviert. Weitere Informationen erhalten Sie unter the section called “Evaluieren Sie CIS Docker Benchmark-Checks” und the section called “Zeichnen Sie die FIPS-Konfigurationsindikatoren auf”.

Installation Sbomgen

Sbomgenist nur für Linux-Betriebssysteme verfügbar.

Sie müssen Docker installiert sein, wenn Sie lokal zwischengespeicherte Bilder analysieren möchtenSbomgen. Dockerist nicht erforderlich, um Bilder zu analysieren, die als `.tar` Dateien exportiert wurden, oder Bilder, die in Remote-Container-Registern gehostet werden.

Amazon Inspector empfiehlt, dass Sie von einem System Sbomgen aus arbeiten, das mindestens die folgenden Hardwarespezifikationen aufweist:

- 4-fache Kern-CPU
- 8 GB RAM

Um zu installieren Sbomgen

1. Laden Sie die neueste Sbomgen ZIP-Datei von der richtigen URL für Ihre Architektur herunter:

   Linux AMD64: https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/amd64/inspector-sbomgen.zip

Linux ARM64: https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/arm64/inspector-sbomgen.zip

Alternativ können Sie frühere Versionen der Amazon Inspector SBOM Generator-Zip-Datei herunterladen.

2. Entpacken Sie den Download mit dem folgenden Befehl:

```
unzip inspector-sbomgen.zip
```

3. Suchen Sie im entpackten Verzeichnis nach den folgenden Dateien:

   - `inspector-sbomgen`— Dies ist das Tool, das Sie ausführen werden, um SBOMs zu generieren.
   - `README.txt`— Dies ist die Dokumentation zur VerwendungSbomgen.
   - `LICENSE.txt`— Diese Datei enthält die Softwarelizenz fürSbomgen.
   - `licenses`— Dieser Ordner enthält Lizenzinformationen für Pakete von Drittanbietern, die von verwendet werdenSbomgen.
   - `checksums.txt`— Diese Datei enthält Hashes des Sbomgen Tools.
   - `sbom.json`— Dies ist eine CycloneDX SBOM für das Tool. Sbomgen
   - `WhatsNew.txt`— Diese Datei enthält ein zusammengefasstes Änderungsprotokoll, sodass Sie wichtige Änderungen und Verbesserungen zwischen den Sbomgen Versionen schnell einsehen können.

4. (Optional) Überprüfen Sie die Authentizität und Integrität des Tools mit dem folgenden Befehl:

```
sha256sum < inspector-sbomgen
```

   - Vergleichen Sie die Ergebnisse mit dem Inhalt der `checksums.txt` Datei.

5. Erteilen Sie dem Tool mit dem folgenden Befehl die Rechte zur ausführbaren Datei:

```
chmod +x inspector-sbomgen
```

6. Stellen Sie mit Sbomgen dem folgenden Befehl sicher, dass die Installation erfolgreich abgeschlossen wurde:

```
./inspector-sbomgen --version
```

Sie sollten die Ausgabe ähnlich der folgenden sehen:

```
Version: 1.X.X
```

Verwenden Sbomgen

In diesem Abschnitt werden verschiedene Möglichkeiten beschrieben, die Sie verwenden können. Sbomgen Anhand der integrierten Beispiele können Sie mehr Sbomgen über die Verwendung erfahren. Führen Sie den folgenden `list-examples` Befehl aus, um sich diese Beispiele anzusehen:

```
./inspector-sbomgen list-examples
```

Generieren Sie eine SBOM für ein Container-Image und geben Sie das Ergebnis aus

Sie können Sbomgen damit SBOMs für Container-Images generieren und das Ergebnis in eine Datei ausgeben. Diese Funktion kann mit dem `container` Unterbefehl aktiviert werden.

Beispielbefehl

Im folgenden Codeausschnitt können Sie es durch die ID Ihres Bilds und *`image:tag output_path.json`* durch den Pfad zu der Ausgabe, die Sie speichern möchten, ersetzen.

```
# generate SBOM for container image
./inspector-sbomgen container --image image:tag -o output_path.json
```

> **Note**
>
> Die Dauer und Leistung des Scans hängen von der Bildgröße und der geringen Anzahl der Ebenen ab. Kleinere Bilder verbessern nicht nur die Sbomgen Leistung, sondern reduzieren auch die potenzielle Angriffsfläche. Kleinere Images verbessern auch die Dauer der Erstellung, des Herunterladens und Uploads von Images.

Bei Verwendung Sbomgen mit ScanSbomverarbeitet die Amazon Inspector Scan API keine SBOMs, die mehr als 5.000 Pakete enthalten. In diesem Szenario gibt die Amazon Inspector Scan API eine HTTP 400-Antwort zurück.

Wenn ein Bild Massenmediendateien oder -verzeichnisse enthält, sollten Sie erwägen, sie von der Sbomgen Verwendung des `--skip-files` Arguments auszuschließen.

Beispiel: Häufige Fehlerfälle

Das Scannen von Container-Images kann aufgrund der folgenden Fehler fehlschlagen:

- `InvalidImageFormat`— Tritt auf, wenn falsch formatierte Container-Images mit beschädigten TAR-Headern, Manifestdateien oder Konfigurationsdateien gescannt werden.
- `ImageValidationFailure`— Tritt auf, wenn die Überprüfung der Prüfsumme oder der Inhaltslänge für Container-Image-Komponenten fehlschlägt, wie z. B. nicht übereinstimmende Content-Length Header, falsche Manifest-Digests oder eine fehlgeschlagene SHA256-Prüfsummenüberprüfung.
- `ErrUnsupportedMediaType`— Tritt auf, wenn Bildkomponenten Medientypen enthalten, die nicht unterstützt werden. Informationen zu unterstützten Medientypen finden Sie unter Unterstützte Betriebssysteme und Medientypen.

Amazon Inspector unterstützt den `application/vnd.docker.distribution.manifest.list.v2+json` Medientyp nicht. Amazon Inspector unterstützt jedoch Manifestlisten. Beim Scannen von Bildern, die Manifestlisten verwenden, können Sie mit dem `--platform` Argument explizit angeben, welche Plattform verwendet werden soll. Wenn das `--platform` Argument nicht angegeben ist, wählt der Amazon Inspector SBOM Generator das Manifest automatisch basierend auf der Plattform aus, auf der es ausgeführt wird.

Generieren Sie eine SBOM aus Verzeichnissen und Archiven

Sie können sie verwendenSbomgen, um SBOMs aus Verzeichnissen und Archiven zu generieren. Diese Funktion kann mit den `archive` Unterbefehlen `directory` oder aktiviert werden. Amazon Inspector empfiehlt, diese Funktion zu verwenden, wenn Sie eine SBOM aus einem Projektordner, z. B. einem heruntergeladenen Git-Repository, generieren möchten.

Beispielbefehl 1

Der folgende Ausschnitt zeigt einen Unterbefehl, der eine SBOM aus einer Verzeichnisdatei generiert.

```
# generate SBOM from directory
./inspector-sbomgen directory --path /path/to/dir -o /tmp/sbom.json
```

Beispiel für Befehl 2

Der folgende Ausschnitt zeigt einen Unterbefehl, der eine SBOM aus einer Archivdatei generiert. Die einzigen unterstützten Archivformate sind, und. `.zip` `.tar` `.tar.gz`

```
# generate SBOM from archive file (tar, tar.gz, and zip formats only)
./inspector-sbomgen archive --path testData.zip -o /tmp/sbom.json
```

Generieren Sie eine SBOM aus Go oder Rust kompilierte Binärdateien

Sie können es verwendenSbomgen, um SBOMs aus kompilierten Dateien Go und Rust Binärdateien zu generieren. Sie können diese Fähigkeit mit dem folgenden Unterbefehl aktivieren: `binary`

```
./inspector-sbomgen binary --path /path/to/your/binary
```

Generieren Sie eine SBOM aus bereitgestellten Volumes

Sie können Amazon Inspector SBOM Generator verwenden, um SBOMs aus bereitgestellten Volumes zu generieren. Diese Funktion kann mit dem Unterbefehl aktiviert werden. `volume` Wir empfehlen, diese Funktion zu verwenden, wenn Sie Speichervolumes analysieren möchten, z. B. Amazon EBS-Volumes, die auf Ihrem System bereitgestellt wurden. Im Gegensatz zum Unterbefehl directory werden beim Scannen von Mounted Volumes Betriebssystempakete und Betriebssysteminformationen erkannt.

Sie können ein Amazon EBS-Volume scannen, indem Sie es an eine Amazon EC2 EC2-Instance anhängen, auf der Amazon Inspector SBOM Generator installiert ist, und es auf dieser Instance mounten. Für Amazon EBS-Volumes, die derzeit von anderen Amazon EC2 EC2-Instances verwendet werden, können Sie einen Amazon EBS-Snapshot des Volumes erstellen und dann aus diesem Snapshot ein neues Amazon EBS-Volume für Scanzwecke erstellen. Weitere Informationen zu Amazon EBS finden Sie unter Was ist Amazon EBS? im Amazon Elastic Block Store-Benutzerhandbuch.

Beispielbefehl

Der folgende Ausschnitt zeigt einen Unterbefehl, der aus einem bereitgestellten Volume eine SBOM generiert. Das `--path` Argument sollte das Stammverzeichnis angeben, in dem das Volume eingehängt ist.

```
# generate SBOM from mounted volume
```

```
./inspector-sbomgen volume --path /mount/point/of/volume/root
```

Beispielbefehl

Der folgende Ausschnitt zeigt einen Unterbefehl, der eine SBOM aus einem bereitgestellten Volume generiert und dabei bestimmte Dateipfade mit dem Argument ausschließt. `--exclude-suffix` Das `--exclude-suffix` Argument ist besonders nützlich, wenn ein Volume Massendateien (wie Protokolldateien oder Mediendateien) enthält. Dateien und Verzeichnisse, deren Pfade mit den angegebenen Suffixen enden, werden vom Scannen ausgeschlossen, wodurch die Scanzeit und der Speicherverbrauch reduziert werden können.

```
# generate SBOM from mounted volume with exclusions
./inspector-sbomgen volume --path /mount/point/of/volume/root \
--exclude-suffix .log \
--exclude-suffix cache
```

Alle Dateipfade im Zielvolume werden auf ihre ursprünglichen Pfade normalisiert. Wenn beispielsweise ein Volume gescannt wird, auf dem `/mnt/volume` sich eine Datei befindet`/mnt/volume/var/lib/rpm/rpmdb.sqlite`, wird der Pfad `/var/lib/rpm/rpmdb.sqlite` in der generierten SBOM auf den Pfad normalisiert.

Senden Sie eine SBOM zur Identifizierung von Sicherheitslücken an Amazon Inspector

Sie können nicht nur eine SBOM generieren, sondern auch eine SBOM zum Scannen mit einem einzigen Befehl aus der Amazon Inspector Scan API senden. Amazon Inspector bewertet den Inhalt der SBOM auf Sicherheitslücken, bevor die Ergebnisse an zurückgegeben werden. Sbomgen Abhängig von Ihren Eingaben können die Ergebnisse angezeigt oder in eine Datei geschrieben werden.

> **Note**
>
> Sie müssen über einen aktiven Benutzer AWS-Konto mit Leseberechtigungen verfügen, `InspectorScan-ScanSbom` um diese Funktion nutzen zu können.

Um diese Funktion zu aktivieren, übergeben Sie das `--scan-sbom` Argument an die Sbomgen CLI. Sie können das `--scan-sbom` Argument auch an einen der folgenden Sbomgen Unterbefehle übergeben:`archive`,,`binary`, `containerdirectory`,`localhost`.

> **Note**
>
> Die Amazon Inspector Scan API verarbeitet keine SBOMs mit mehr als 5.000 Paketen. In diesem Szenario gibt die Amazon Inspector Scan API eine HTTP 400-Antwort zurück.

Sie können sich bei Amazon Inspector über ein AWS Profil oder eine IAM-Rolle mit den folgenden AWS CLI Argumenten authentifizieren:

```
--aws-profile profile
--aws-region region
--aws-iam-role-arn role_arn
```

Sie können sich auch bei Amazon Inspector authentifizieren, indem Sie die folgenden Umgebungsvariablen angeben. Sbomgen

```
AWS_ACCESS_KEY_ID=$access_key \
AWS_SECRET_ACCESS_KEY=$secret_key \
AWS_DEFAULT_REGION=$region \
./inspector-sbomgen arguments
```

Um das Antwortformat anzugeben, verwenden Sie das `--scan-sbom-output-format cyclonedx` Argument oder das `--scan-sbom-output-format inspector` Argument.

Beispiel für Befehl 1

Dieser Befehl erstellt eine SBOM für die neueste Alpine Linux Version, scannt die SBOM und schreibt die Ergebnisse der Sicherheitslücke in eine JSON-Datei.

```
./inspector-sbomgen container --image alpine:latest \
                          --scan-sbom \
                          --aws-profile your_profile \
                          --aws-region your_region \
                          --scan-sbom-output-format cyclonedx \
                          --outfile /tmp/inspector_scan.json
```

Beispielbefehl 2

Dieser Befehl authentifiziert sich bei Amazon Inspector und verwendet AWS Anmeldeinformationen als Umgebungsvariablen.

```
AWS_ACCESS_KEY_ID=$your_access_key \
AWS_SECRET_ACCESS_KEY=$your_secret_key \
AWS_DEFAULT_REGION=$your_region \
./inspector-sbomgen container --image alpine:latest \
                          -o /tmp/sbom.json \
                          --scan-sbom \
                          --scan-sbom-output-format inspector
```

Beispielbefehl 3

Dieser Befehl authentifiziert sich bei Amazon Inspector mithilfe des ARN für eine IAM-Rolle.

```
./inspector-sbomgen container --image alpine:latest \
                          --scan-sbom \
                          --aws-profile your_profile \
                          --aws-region your_region \
                          --outfile /tmp/inspector_scan.json
                          --aws-iam-role-arn arn:aws:iam::123456789012:role/your_role
```

Verwenden Sie zusätzliche Scanner, um die Erkennungsfunktionen zu verbessern

Der Amazon Inspector SBOM Generator wendet vordefinierte Scanner auf der Grundlage des verwendeten Befehls an.

Standard-Scanner-Gruppen

Jeder Amazon Inspector SBOM Generator-Unterbefehl wendet die folgenden Standard-Scanner-Gruppen automatisch an.

- Für den `directory` Unterbefehl: Binary, Programming Language-Packages, Docker-File-Scanner-Gruppen
- Für den `localhost` Unterbefehl: os, programing-language-packages, Scanner-Gruppen für Extra-Ökosysteme
- Für den `container` Unterbefehl: os, programing-language-packages, extra-ecosystems, dockerfile, binary scanner groups

Spezielle Scanner

Verwenden Sie die `--additional-scanners` Option, gefolgt vom Namen des Scanners, der hinzugefügt werden soll, um andere Scanner als die Standard-Scanner-Gruppen einzubeziehen. Im Folgenden finden Sie einen Beispielbefehl, der zeigt, wie das geht.

```
# Add WordPress installation scanner to directory scan
./inspector-sbomgen directory --path /path/to/directory/ --additional-scanners
 wordpress-installation -o output.json
```

Im Folgenden finden Sie einen Beispielbefehl, der zeigt, wie Sie mehrere Scanner mit einer durch Kommas getrennten Liste hinzufügen.

```
./inspector-sbomgen container --image image:tag --additional-scanners scanner1,scanner2
 -o output.json
```

Optimieren Sie Containerscans, indem Sie die maximale zu scannende Dateigröße anpassen

Wenn Sie ein Container-Image analysieren und verarbeiten, werden standardmäßig Dateien mit einer Größe von 200 MB oder weniger Sbomgen gescannt. Dateien, die größer als 200 MB sind, enthalten selten Paketmetadaten. Bei der Inventarisierung von Dateien Go oder Rust Binärdateien mit mehr als 200 MB können Fehler auftreten. Verwenden Sie das `--max-file-size` Argument, um die Größenbeschränkung anzupassen. Auf diese Weise können Sie das Limit erhöhen, um große Dateien einzuschließen, und das Limit verringern, um den Ressourcenverbrauch zu reduzieren, indem Sie große Dateien ausschließen.

Beispiel

Das folgende Beispiel zeigt, wie das `--max-file-size` Argument verwendet wird, um die Dateigröße zu erhöhen.

```
# Increase the file size limit to scan files up to 300 MB
./inspector-sbomgen container --image alpine:latest \
--outfile /tmp/sbom.json \
```

```
--max-file-size 300000000
```

Durch das Anpassen dieser Einstellung können Sie die Festplattennutzung, den Speicherverbrauch und die Gesamtdauer des Scans kontrollieren.

Beschränken Sie die Abrufgröße für Container-Images

Wenn Sie ein Container-Image scannen, können Sie ein übergroßes Bild zurückweisen, bevor Sie eine Ebene Sbomgen herunterladen. Sbomgenliest die komprimierte Größe jeder Ebene aus dem Image-Manifest und vergleicht sie mit den von Ihnen festgelegten Grenzwerten, sodass ein Bild, das einen Grenzwert überschreitet, zurückgewiesen wird, ohne dass seine Ebenen heruntergeladen werden. Verwenden Sie die `--layer-max-pull-size` Argumente `--image-max-pull-size` und, um diese Grenzwerte in Byte festzulegen.

- `--image-max-pull-size`— Weist das Bild ab, wenn die komprimierten Größen aller Ebenen zusammen mehr als die angegebene Bytezahl ergeben.
- `--layer-max-pull-size`— Weist das Bild ab, wenn die komprimierte Größe einer einzelnen Ebene die angegebene Byteanzahl überschreitet.

> **Note**
>
> Beide Argumente gelten nur für Containerscans und sind unabhängig voneinander. Jedes Feld ist standardmäßig auf voreingestellt`0`, wodurch das Limit deaktiviert wird.

Standardmäßig führt eine Ebene, deren komprimierte Größe den Wert überschreitet, `--layer-max-pull-size` Sbomgen dazu, dass das gesamte Bild zurückgewiesen wird. Um die übergroße Ebene zu überspringen und stattdessen mit der Inventarisierung des restlichen Bilds fortzufahren, fügen Sie das Argument hinzu. `--continue-on-layer-size-exceeded` Es hat keine Wirkung, sofern es nicht gesetzt `--layer-max-pull-size` ist, und lehnt ein `--image-max-pull-size` übergroßes Bild trotzdem ab.

Jede übersprungene Ebene wird als `data` Unterkomponente der Bild-Metadaten-Komponente gemeldet, benannt nach ihrer Diff-ID, mit den folgenden Eigenschaften:

- `amazon:inspector:sbom_generator:layer:skip_reason`— Warum die Ebene übersprungen wurde (). `layer_size_exceeded`

- `amazon:inspector:sbom_generator:layer:size_bytes`— Die Größe der übersprungenen Ebene.
- `amazon:inspector:sbom_generator:layer:size_limit_bytes`— Das Limit, das die Ebene überschritten hat.

> **Note**
>
> Die Whiteouts einer übersprungenen Ebene werden nicht angewendet. Ein Pfad, den die Ebene löscht oder ersetzt, wird also so gemeldet, wie die Ebene darunter ihn verlassen hat.

Beispiel

Im folgenden Beispiel wird das Bild zurückgewiesen, wenn die komprimierte Größe seiner Ebenen insgesamt mehr als 50 GiB (`53687091200`Byte) beträgt oder wenn die komprimierte Größe einer einzelnen Ebene 10 GiB (`10737418240`Byte) überschreitet.

```
# Reject an oversized image before downloading any layer
./inspector-sbomgen container --image image:tag \
--outfile /tmp/sbom.json \
--image-max-pull-size 53687091200 \
--layer-max-pull-size 10737418240
```

Beispiel

Im folgenden Beispiel werden alle Ebenen, die größer als 10 GiB (`10737418240`Byte) sind, übersprungen und die verbleibenden Ebenen inventarisiert, anstatt das Bild zurückzuweisen.

```
# Skip oversized layers instead of rejecting the image
./inspector-sbomgen container --image image:tag \
--outfile /tmp/sbom.json \
--layer-max-pull-size 10737418240 \
--continue-on-layer-size-exceeded
```

Erkennt Scratch-Bilder mit Dateisystem-Heuristiken

Wenn Sbomgen Sie das Betriebssystem eines Container-Images nicht identifizieren können, können Sie das `--enable-scratch-heuristics` Argument verwenden, um das Dateisystem auf Signale zu untersuchen, die auf ein Bild hinweisen. scratch Wenn mindestens zwei Signale vorhanden sind,

wird anstelle von die Betriebssystemkomponente Sbomgen beschriftet. `scratch` unknown Ein identifiziertes Betriebssystem wird niemals neu etikettiert.

> **Note**
>
> Dieses Argument ist experimentell, gilt nur für Containerscans und ist standardmäßig deaktiviert.

Sbomgenwertet die folgenden Signale aus und zeichnet jedes erkannte Signal als `amazon:inspector:sbom_generator:scratch:heuristic:*` Eigenschaft der Betriebssystemkomponente auf:

- `missing_lib_dir`— Das Verzeichnis fehlt`/lib`.
- `missing_bin_dir`— Das `/bin` Verzeichnis fehlt.
- `missing_var_dir`— Das `/var` Verzeichnis fehlt.
- `missing_shell`— Es ist keine interaktive Shell vorhanden (keine von`sh`,`bash`,`ash`, `dashzsh`, oder `busybox` wurde unter `/bin` oder gefunden`/usr/bin`).

Beispiel

Das folgende Beispiel zeigt, wie das `--enable-scratch-heuristics` Argument verwendet wird.

```
# enable scratch image detection heuristics
./inspector-sbomgen container --image image:tag \
--outfile /tmp/sbom.json \
--enable-scratch-heuristics
```

> **Note**
>
> Dieses Verhalten ist heuristisch und kann zu Fehlalarmen führen. Es ändert nur die Betriebssystemkomponente.

Sammeln Sie Metadaten für laufende Prozesse

Wenn Sie einen Localhost scannen, können Sie das `--collect-processes` Argument verwenden, um Komponenten mit Metadaten zu den laufenden Prozessen zu kennzeichnen, die Sbomgen vom

Host gesammelt werden. Auf diese Weise können Sie eine inventarisierte Komponente mit den Prozessen korrelieren, die auf ihr ausgeführt werden.

> **Note**
>
> Dieses Argument ist Opt-In, standardmäßig deaktiviert und gilt nur für Localhost-Scans. Wenn Sie es für einen anderen Scantyp angeben, wird eine Warnung Sbomgen protokolliert und ignoriert.

Wenn eine Komponente mit einem laufenden Prozess übereinstimmt, Sbomgen fügt dieser Komponente die folgenden Eigenschaften hinzu, wobei die Prozess-ID angegeben *`pid`* ist:

- `amazon:inspector:sbom_generator:host:process_id:`*`pid`*`:path`— Zum Beispiel der Pfad der ausführbaren Datei des entsprechenden laufenden Prozesses`/usr/sbin/sshd`.
- `amazon:inspector:sbom_generator:host:process_id:`*`pid`*`:started_at`— Die Uhrzeit, zu der der Prozess gestartet wurde, zum Beispiel im RFC 3339-Format (UTC). `2026-08-26T15:43:40Z`

Beispiel

Das folgende Beispiel zeigt, wie das Argument verwendet wird`--collect-processes`.

```
# Tag components with running process metadata
./inspector-sbomgen localhost \
--outfile /tmp/sbom.json \
--collect-processes
```

Evaluieren Sie CIS Docker Benchmark-Checks

Wenn Sie ein Container-Image scannen, können Sie das `--enable-cis-docker` Argument verwenden, um unterstützte CIS Docker Benchmark Prüfungen auszuwerten und das Ergebnis jeder Prüfung als Eigenschaft der Container-Komponente aufzuzeichnen.

> **Note**
>
> Dieses Argument ist experimentell, gilt nur für Containerscans und ist standardmäßig deaktiviert.

Sbomgenzeichnet die folgenden Eigenschaften der Container-Komponente auf, wobei *check* es sich um den Bezeichner einer einzelnen Prüfung handelt:

- `amazon:inspector:sbom_generator:image:cis:docker:`*check*— Das `pass` oder `fail` Ergebnis einer einzelnen Prüfung, eine pro unterstützter Prüfung (derzeit 4.1, 4.6, 4.7, 4.9 und 4.10).
- `amazon:inspector:sbom_generator:image:cis:docker:checks_passed`— Die Anzahl der bewerteten Prüfungen, die bestanden wurden.

Beispiel

Das folgende Beispiel zeigt, wie das `--enable-cis-docker` Argument verwendet wird.

```
# Evaluate CIS Docker Benchmark checks
./inspector-sbomgen container --image image:tag \
--outfile /tmp/sbom.json \
--enable-cis-docker
```

Zeichnen Sie die FIPS-Konfigurationsindikatoren auf

Wenn Sie ein Linux Container-Image oder einen Localhost scannen, können Sie das `--enable-fips-checks` Argument verwenden, um Indikatoren aufzuzeichnen, die die FIPS-related kryptografische Konfiguration des Systems beschreiben.

> **Note**
>
> Dieses Argument ist experimentell, gilt nur für Linux Container- und Localhost-Scans und ist standardmäßig deaktiviert.

Sbomgenzeichnet die folgenden Eigenschaften mit jeweils einem Wert von `passfail`, oder `unknown` auf:

- `amazon:inspector:sbom_generator:fips:aide_fips_hashes`
- `amazon:inspector:sbom_generator:fips:boot_fips_arg`
- `amazon:inspector:sbom_generator:fips:crypto_policy_fips`
- `amazon:inspector:sbom_generator:fips:dracut_fips_module`
- `amazon:inspector:sbom_generator:fips:kernel_fips_enabled`

- `amazon:inspector:sbom_generator:fips:os_fips_certified`
- `amazon:inspector:sbom_generator:fips:stig_crypto_subpolicy`

Beispiel

Das folgende Beispiel zeigt, wie das `--enable-fips-checks` Argument bei einem Localhost-Scan verwendet wird.

```
# Record FIPS configuration indicators
./inspector-sbomgen localhost \
--outfile /tmp/sbom.json \
--enable-fips-checks
```

Deaktivieren Sie die Fortschrittsanzeige

Sbomgenzeigt eine sich drehende Fortschrittsanzeige an, die in CI/CD Umgebungen zu viele Schrägstriche führen kann.

```
INFO[2024-02-01 14:58:46]coreV1.go:53: analyzing artifact
|
\
/
|
\
/
INFO[2024-02-01 14:58:46]coreV1.go:62: executing post-processors
```

Sie können die Fortschrittsanzeige mit dem folgenden `--disable-progress-bar` Argument deaktivieren:

```
./inspector-sbomgen container --image alpine:latest \
--outfile /tmp/sbom.json \
--disable-progress-bar
```

Authentifizierung bei privaten Registern mit Sbomgen

Indem Sie Ihre Anmeldedaten für die private Registrierung angeben, können Sie SBOMs aus Containern generieren, die in privaten Registern gehostet werden. Sie können diese Anmeldeinformationen mit den folgenden Methoden bereitstellen:

Authentifizieren Sie sich mit zwischengespeicherten Anmeldeinformationen (empfohlen)

Für diese Methode authentifizieren Sie sich bei Ihrer Container-Registry. Wenn Sie dies beispielsweise verwendenDocker, können Sie sich mit dem Docker Loging-Befehl bei Ihrer Container-Registry authentifizieren:. `docker login`

1. Authentifizieren Sie sich bei Ihrer Container-Registry. Wenn Sie dies beispielsweise verwendenDocker, können Sie sich mit dem folgenden Befehl bei Ihrer Registrierung authentifizieren: Docker `login`
2. Nachdem Sie sich bei Ihrer Container-Registry authentifiziert haben, verwenden Sie es Sbomgen auf einem Container-Image, das sich in der Registrierung befindet. Um das folgende Beispiel zu verwenden, *`image:tag`* ersetzen Sie es durch den Namen des zu scannenden Bilds:

```
./inspector-sbomgen container --image image:tag
```

Authentifizieren Sie sich mit der interaktiven Methode

Geben Sie für diese Methode Ihren Benutzernamen als Parameter an und Sie Sbomgen werden bei Bedarf zur sicheren Passworteingabe aufgefordert.

Um das folgende Beispiel zu verwenden, *`image:tag`* ersetzen Sie es durch den Namen des Bilds, das Sie scannen möchten, und *`your_username`* durch einen Benutzernamen, der Zugriff auf das Bild hat:

```
./inspector-sbomgen container --image image:tag --username your_username
```

Authentifizieren Sie sich mit der nicht interaktiven Methode

Speichern Sie für diese Methode Ihr Passwort oder Ihr Registrierungstoken in einer `.txt` Datei.

> **Note**
>
> Der aktuelle Benutzer sollte diese Datei nur lesen können. Die Datei sollte auch Ihr Passwort oder Token in einer einzigen Zeile enthalten.

Um das folgende Beispiel zu verwenden, *your_username* ersetzen Sie es durch Ihren Benutzernamen, *password.txt* durch die `.txt` Datei, die Ihr Passwort oder Token in einer einzigen Zeile enthält, und *image:tag* durch den Namen des zu scannenden Bilds:

```
INSPECTOR_SBOMGEN_USERNAME=your_username \
INSPECTOR_SBOMGEN_PASSWORD=`cat password.txt` \
./inspector-sbomgen container --image image:tag
```

Beispielausgaben von Sbomgen

Im Folgenden finden Sie ein Beispiel für eine SBOM für ein Container-Image, das mit Hilfe von inventarisiert wurde. Sbomgen

Container-Image (SBOM)

```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.5",
  "serialNumber": "urn:uuid:828875ef-8c32-4777-b688-0af96f3cf619",
  "version": 1,
  "metadata": {
    "timestamp": "2023-11-17T21:36:38Z",
    "tools": [
      {
        "vendor": "Amazon Web Services, Inc. (AWS)",
        "name": "Amazon Inspector SBOM Generator",
        "version": "1.0.0",
        "hashes": [
          {
            "alg": "SHA-256",
            "content":
 "10ab669cfc99774786301a745165b5957c92ed9562d19972fbf344d4393b5eb1"
          }
        ]
      }
    ],
    "component": {
      "bom-ref": "comp-1",
      "type": "container",
      "name": "fedora:latest",
      "properties": [
        {
```

```
          "name": "amazon:inspector:sbom_generator:image_id",
          "value":
 "sha256:c81c8ae4dda7dedc0711daefe4076d33a88a69a28c398688090c1141eff17e50"
        },
        {
          "name": "amazon:inspector:sbom_generator:layer_diff_id",
          "value":
 "sha256:eddd0d48c295dc168d0710f70364581bd84b1dda6bb386c4a4de0b61de2f2119"
        }
      ]
    }
  },
  "components": [
    {
      "bom-ref": "comp-2",
      "type": "library",
      "name": "dnf",
      "version": "4.18.0",
      "purl": "pkg:pypi/dnf@4.18.0",
      "properties": [
        {
          "name": "amazon:inspector:sbom_generator:source_file_scanner",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_package_collector",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_path",
          "value": "/usr/lib/python3.12/site-packages/dnf-4.18.0.dist-info/METADATA"
        },
        {
          "name": "amazon:inspector:sbom_generator:is_duplicate_package",
          "value": "true"
        },
        {
          "name": "amazon:inspector:sbom_generator:duplicate_purl",
          "value": "pkg:rpm/fedora/python3-dnf@4.18.0-2.fc39?
arch=noarch&distro=39&epoch=0"
        }
      ]
    },
    {
```

```
      "bom-ref": "comp-3",
      "type": "library",
      "name": "libcomps",
      "version": "0.1.20",
      "purl": "pkg:pypi/libcomps@0.1.20",
      "properties": [
        {
          "name": "amazon:inspector:sbom_generator:source_file_scanner",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_package_collector",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_path",
          "value": "/usr/lib64/python3.12/site-packages/libcomps-0.1.20-py3.12.egg-
info/PKG-INFO"
        },
        {
          "name": "amazon:inspector:sbom_generator:is_duplicate_package",
          "value": "true"
        },
        {
          "name": "amazon:inspector:sbom_generator:duplicate_purl",
          "value": "pkg:rpm/fedora/python3-libcomps@0.1.20-1.fc39?
arch=x86_64&distro=39&epoch=0"
        }
      ]
    }
  ]
}
```


Frühere Versionen des Amazon Inspector SBOM Generators

Dieses Thema enthält Links zu den neuesten und früheren Versionen des Amazon Inspector SBOM Generators. Informationen zur Installation von Sbomgen finden Sie im Abschnitt Installieren der Sbomgen.

Aktuelle Version

- https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/amd64/inspector-sbomgen.zip

- https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/arm64/inspector-sbomgen.zip

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.17.2 1.17.2	d0ddc22ef8428a4f4ab8eb6d6feca0d3d24b1c3b65cce3a5f1845c5234fe5428 2fd1cf4919948e77a6cbde320a176ff7b5a25f439d176a3dae7955d31ab794fa
Linux AMD64 Linux ARM64	1.17.1 1.17.1	e1c11ad4a9752ec19d836f9a67664f701aa0af83f8a5a7e786593d34fb077a80 71d3704c508b654d7ccf8b50ab09980398c59ac4b32b9a506be2571198337a9c
Linux AMD64 Linux ARM64	1.17.0 1,17.0	bec950646a347f6637615d5d6d240c8490fb2f1e8fe16b894527ca612db6cc5e 7e10efc9cbbf5e6965c3f2df3ed8283d3fff4938019d67f0d2104d15cb883230
Linux AMD64 Linux ARM64	1.16.0 1.16.0	2aff31bd5f7f426020a5cdda7a58aaebde72e7dae9bc78f687978ad19ef487b4

Plattform	Version	SHA-256 Prüfsumme
		c0ee096fe6e25123b8420bdd09a14d0dbd15333c017825c6cc815ce68e465d0d
Linux AMD64 Linux ARM64	1.15.5 1,1,5	6dd27c02e9071aff17d41331edc7d4da74324c3b2354f85dfcb630079715662f 4098a459676f32a6e43799028e2cc674b558ac4384fa845947cc6740571a6f3c
Linux AMD64 Linux ARM64	1.15.4 1.15,4	d379121b4b117c2c5e0ddc7b9a964fd9b1366a02b0924f93bb36d5e5edb04842 05263f81ce46e177e17a6e9b9a477f4d03d5129aabe4351d3e509cbf0c6531a5
Linux AMD64 Linux ARM64	1.15.3 1.15,3	3b0a8ebb82c1ccf683b06794cc30b7106185f4c9c6e527d98c8f2386cb9f3303 ed0f2c796b0ac2a8f52d2c8f5a0722f7a72ed2dca3811cc726cbe9c29630c915

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.15.2 1.15,2	fdab1852463cc7590333ff3d0bb6c088f0da01cbce287c97a2c448e183efa643 96b87405d56f48d8697b6da2bbe8a868b56bfc053c4278afc2456e9ade810f9f
Linux AMD64 Linux ARM64	1.15.1 1.15,1	405d527933943e56b884773b9871b1a63beb5ee06a2457e7dd6ead6c69217737 d57d203d1b1085968b4ad9109ad638a8b069fa15d3d0bc008eadba8350df7a0c
Linux AMD64 Linux ARM64	1.15.0 1.15,0	bdd29a252820265dc8715ea69bd15e0cbd1716397f923e3f70fd746775ee82ee e85614cd3343f0233190e85981bb5ea18482214a1d218cb2287c3b4e025a06eb

Plattform	Version	SHA-256 Prüfsumme	
Linux AMD64 Linux ARM64	1.14.4 1.14.4	336d8aad1d57c41c67bc1f0252579f29fa16d71d6d8a5c5908e04599bcaf28f3 4389ed3df5cfd0817d35a6284d93bc6457c248826b723240f92e3826a32b7e97	
Linux AMD64 Linux ARM64	1.14.3 1.14,3	a498d19194c4c79c1efc8a69f6e512ee9e86290cff7dc7fb45bd4b647e5fa629 d06b822e633d9d80dc81597d1c17f6df424b542c62dfd05671bc39fd1c7c0f95	
Linux AMD64 Linux ARM64	1.14.2 1.14.2	26f019f09dc5f4ce7389e9cf2a852bb2ab5c918740ce13dff5ca68bccea20965 fca50fcee81f14272c1f861ee8d5507d0fdcc67d555056ae78ec18d0707a7966	

Plattform	Version	SHA-256 Prüfsumme	
Linux AMD64 Linux ARM64	1.14.1 1.14.1	2b9512ea9acdc1eb1a a155f7f664f92b4dc9 6e14d3849fe1f1d3aa 51f481b627 f23dafdf0127dfd61a f2678aaa67643bf1f0 909689487febe68143 0302a0c8a3	
Linux AMD64 Linux ARM64	1.14.0 1.14,0	f884433321762afec8 89acb78b935a9a6f5f b92fd4bbfa1a54ac03 7e9f24c351 55c62f54dd96e13c08 5acf946cd2a841453a f00918050331a86b2b 83d1f80eaa	
Linux AMD64 Linux ARM64	1.13.2 1.13,2	86011ffc7ee1a85d4c b411ad807566d7fde3 7fe9756a9a7e5a2d22 205645f2d2 d4b6328867f0cac169 8ec4e0862cbdf45af0 68888ce3c1ed30d780 b110c909af	

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.13.1 1.13.1	c100ff7a65f2795ada9193aea0f2944ec67731313a6eb8b841a0ffbf0ebba2c9 3ceaac63796f6ca97e9b5ae7c29bf8c896b20b2616bacde5ea12655c30467164
Linux AMD64 Linux ARM64	1.13.0 1.13,0	f6d81b76f46a7461a2191a72880a7ee90a13ebd4c1b1a2ab6cbc7b469613bd93 6d69c2394df58ccf0940f3495044b90c3f5feca2650d7af14f2000a537c4e96e
Linux AMD64 Linux ARM64	1.12.2 1.12.2	c1d77171941f324519bccf78339e71a8961b7bc2a26351772c09dc6fb9dde83f d4b0eee17c269fcaef7361f714976d31cb494c68eadf9a9cd8e628ee14810a7e

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.12.1 1.12.1	2d698b47d 2a6ba0b431c1161bc1 fba7bef4d9041546e6 24086d4b0dfff122a3f c76fe1bd84ffe92709 21df1d708fc0e49c53 aa7717d29a90d5cec3 cee772f0c2
Linux AMD64 Linux ARM64	1.12.0 1.12.0	1112b3952 81ea5d67e 20062429bf1603ba64 56746d6941f2edfa0e e0812b34b6 25e618000ffc4aceaf 002c2fa3e97d0564d3 76acc34e9e442cdd76 9f6c805248
Linux AMD64 Linux ARM64	1.11.2 1.11.2	bef68671bc532e4fb5 29500b62d7af836012 3cd967308d41ad0ce8 f43f7762fb 4f11d7037efa443f44 2c4edf7ba28774c4fa 706fb7622e4fba645b b3ad3958c9

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.11.1 1.11.1	809eb7cb80d24fbf6f fdd124438d53a90763 2c222e924913ebd610 44ca949490 057f9e4c9970aeda4b da0685e7e02436fd52 23fbe81cec65138551 c63ed77ba0
Linux AMD64 Linux ARM64	1.11.0 1.11.0	5172a5556cf46f9fbc 5cf1d35bd382919fb6 b41aca1ec938db3a75 530060b0cf c9e2da7b076dc89dc3 9a962a7dd9c7d1fd29 230a4eec7eb95f951d 6a179093d0
Linux AMD64 Linux ARM64	1.10.1 1.10.1	9e33622a7874adfe71 9ab7db75a1e44f4b5f ae3573374068b501c8 9f0accfcfe 78d5a7f800fc26ba86 adab5b634431a91c00 7075e06d6ce46e5068 7d5156184e

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.10.0 1.10.0	0b7a553d7d2d17c40a62f1a11013bc46fa2c3814f407c11130e15af3fe313769 5ce9e315a4f8f90ff5eed7ab058efc8dbff6593d66d3fc455f1c37e882ec6466
Linux AMD64 Linux ARM64	1.9.1 1.9.1	d0ef4c14fec6c42e70ae55b3e44d17d027132947596e8ef861c0efc3c0e5a871 2d8145011c13f5611fc30f4510785d53e98b911717f6dbe69616af4d4b0df61f
Linux AMD64 Linux ARM64	1.9.0 1.9.0	78b377b2730eb15476173e40885454ae191e953663af3e0928dddfb8608f4655 985bdc06d25eccb87c4a81995c8a2d3c78e1c02beea309a620b2de4954767591

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.8.3 1.8.3	54eed5a772f68320f3906bec5920e3a19da904abdace10f985b87859015eef89 febd74a397fb0cdd3356072503f08465ab872d1620d59a2ab7d83bdb076c929d
Linux AMD64 Linux ARM64	1.8.2 1.8.2	2e4e3c754e230046349dd975feb48fa953ea5a2de190cbbc17c1c85043936b5a 449a49e222a2bdffe0353435d7b04b0556b35a391c7b9714ce46d1a5382bc3e2
Linux AMD64 Linux ARM64	1.8.1 1.8.1	9ff7958e298d2b228b0c7617f0a9a873254587fc26aee9826c37273650b389e9 6737584fd2c7d24b56777d02846d1737f47d0121344baea217a3e5368fd98fcc

Plattform	Version	SHA-256 Prüfsumme	
Linux AMD64 Linux ARM64	1.8.0 1.8.0	ef32e7fb4ee0af1e47d6b528b47293fc7127c7a7539f7354e84452626a4c204d 0b82ddc691a517bb8fc6ccd67b80ca566b117a1bb410c05764c9b7e3ba76c510	
Linux AMD64 Linux ARM64	1.7.3 1.7.3	3fba95d44aaea55ad06d3c7635a671662c483474578376d3f11e84474f8de25f 1f4b52e3d80de87b92b563a78bac4a2d898e7af82db5b6791d899d516e97cfbb	
Linux AMD64 Linux ARM64	1.7.2 1.7.2	c44ba9bf1cfeb3ea2d6d0b15d25816800a5045a438474f2f77c390bac41ae4cb d37c5b1605bf82260da0b0f36311c83b1646a4327c3fd8169ba4b3a978470c9c	

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.7.1 1.7.1	b0beb602a 6ae439d4e 307bd99682bc8a419f d7d5e78a278bfc718e b18e00b05e 95ff2d9df2fcd1982d d705df1e763f57a0b4 99b6fe06801e9a8086 9e2e464831
Linux AMD64 Linux ARM64	1.7.0 1.7.0	a6316c2ecd5fde7091 d1099335f45f0e2400 b3977c92ee4d72bd1e b359320e61 9751ba5e5c6c6c0aef 7d29b1c4adbd4088da 3a07bb77eaa7de3f04 aa33ad8562
Linux AMD64 Linux ARM64	1.6.3 1.6.3	b6a309e87 9aaa78d7d 8e224eb5214df5fd41 5244d370885e6c8876 db5a4181d2 59ed0b7eb 7d1eadadb691f058d3 2634a03a856ba03ac2 ddb8cd3599ceb55cb9 a

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.6.2 1.6.2	8d8ba0653 5be614a4d44b1bd74c 66d1fd4874ff9ab788 ad5e23aa5229db9c68 7 2bd7b4a88b9c6b041a 6ff82f7f9bc116b76c f410bf6eb896fc8d68 e717b55f2a
Linux AMD64 Linux ARM64	1.6.1 1.6.1	3e3d62dc794b31d9d2 de1904592cf42f25e9 f42c30eb90cc53385a 60b42f1a63 ad89f670908fb0b48b ca0242f3ac58e7179f 6fabfcc9a2b3fd0e5c 3d79e27539
Linux AMD64 Linux ARM64	1.6.0 1.6.0	ffe671c2c1d1c2142a 4af056d1c179eaffbc 3925f5afaa6f3d655b d495ce5e1c a733c0b00c7225369c 68ad47c57846b4546e 2c9f47580ab98394ba efc765c134

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.5.5 1.5.5	ebcfbe565631de5bc61b1d55d70a2d15b965f628678a2b60cffd01cd0c3443f1 a8e018ceee3a76dd4271f966015c216438b11ee807fcd970753e786baa335b56
Linux AMD64 Linux ARM64	1.5.4 1.5.4	aa8c1ffacc563b87975497f53eddec0b29397a898fac19f4902b8acb7eeb347b c6ba98d441aa88d3d3150449c098cd13ce3baeccee45ad4c9a1326f8bb8f87fc
Linux AMD64 Linux ARM64	1.5.3 1.5.3	d493c23121101c9c3df888e717bf81d7f7b81809754f3492e1ae52f02b089b68 8dfa5c97b3bd45da487706e95d1894290f53b113247bbb89b9fac16dab8184b6

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.5.2 1.5.2	ff6233d7da9f7e963589a0eb8f07bee2ca375360365cb6b6e354585cf1371910 fd31efb6031754b2bc8414d7fe9dd14a067767704145af0559b3500cc437c7ee
Linux AMD64 Linux ARM64	1.5.1 1.5.1	391fcc52117fed79cae6e92a9e225732166a6df2582aa7f6b5230149761f6732 f9bc90d18724f93db0f5ca3b79136adb7b49fa33fa179a5e87b4d512f256b56b
Linux AMD64 Linux ARM64	1.5.0 1.5.0	d7b6cb84053358e462d76488d019140ecd05ad405217a60a96b727fb062880fe 067dcf5c302160a5270f89aed3f941bb0571dcb8a59f75dddb1b7747c2a82ec7

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.4.0 1.4.0	c8ca73761afd742e1d eb98b04eb5714c9c2a 574b652a7 63b18e235 60e66aea24 188d97577 82278653e65605aaf1 86feda104345ba2f9d e438873e568f1ff6204
Linux AMD64 Linux ARM64	1.3.2 1.3.2	57dd5d135 600e84690706cfe958 60e78149988d37cf81 429ce97b9256d179fb 4 91526ecdafc6cc3718 fabe75b2693ace5eff b9c0af3327b484b7f5 a154929997
Linux AMD64 Linux ARM64	1.3.1 1.3.1	097ec83907c459a36d e11c92d016fffd64f1 c33fd4bcbf2af465e0 979b0d9237 aa93a3d402abc4a986 a9ad9d3de8fcca81ee 25a55596ac6dc4502e d1d6819502

Plattform	Version	SHA-256 Prüfsumme	
Linux AMD64 Linux ARM64	1.3.0 1.3.0	21439f92c314daf136832ca6676a65d288768aa69fc6dcd2014a3038b2701eeb 4a41779b0c3b32242eedef288de6c1bf40fda0d4246b32fd0cd8d4e51e58f94b	
Linux AMD64 Linux ARM64	1.2.1 1.2.1	e022e95e59f1790949bca8dbbb6478a5d3fb677ccd45aa4ba30ebd91ae86ad65 824acc5bb5b0210954fe9ab089d9461453a4975d34292cc0c676837c3a7279b4	
Linux AMD64 Linux ARM64	1.2.0 1.2.0	9625b1a8ae1937ca2179c2535a0ffceca934138e0b66feac9ba3e34ffaa22ec5 7f387e560b41571fb52efd9e620bf2b9e3a067ca781e88aaa977b2b8acdebf35	

Plattform	Version	SHA-256 Prüfsumme
Linux AMD64 Linux ARM64	1.1.1 1.1.1	6809b7e46675c66e3af354c53433dc46c4d1ddaf258e05ba15e38e784ea0285e 6361e59fb2448c66c4698ea33979ecaaefc2af4420034aabbbe741242f60dbdd
Linux AMD64 Linux ARM64	1.1.0 1.1.0	f84c8815413d451490b38509950235f88713c0c61c7259a4831934995664bd8f aaffefb5e44195dc55d5fd3289e511720f64c130644cbd58103cf7f36e96f058
Linux AMD64 Linux ARM64	1.0.0 1.0.0	cc126e24962f1a6497cf17679b3e3b73be68963c47e3968a56e73caacf045b5c 5d5bf97a4acfeaaa73ad6c918738188e0c822e475ef37a334e49d77ba907b08a

Umfassende Betriebssystemsammlung von Amazon Inspector SBOM Generator

Der Amazon Inspector SBOM Generator scannt verschiedene Betriebssysteme, um eine robuste und detaillierte Analyse der Systemkomponenten zu gewährleisten. Die Generierung einer

SBOM hilft Ihnen dabei, die Zusammensetzung Ihres Betriebssystems zu verstehen, sodass Sie Sicherheitslücken in vom System verwalteten Paketen identifizieren können. In diesem Thema werden die wichtigsten Funktionen verschiedener Betriebssystem-Paketsammlungen beschrieben, die der Amazon Inspector SBOM Generator unterstützt. Informationen zu den Betriebssystemen, die Amazon Inspector unterstützt, finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.

Unterstützte Betriebssystemartefakte

Der Amazon Inspector SBOM Generator unterstützt die folgenden Betriebssystemartefakte:

Plattform	Binär	Quelle	Streamen
Alma Linux	N/A	Ja	Ja
Alpine Linux	Ja	Ja	N/A
Amazon Linux	N/A	Ja	N/A
CentOS	N/A	Ja	N/A
Chainguard	Ja	Ja	N/A
Debian	Ja	Ja	N/A
Distroless	Ja	Ja	N/A
Fedora	N/A	Ja	N/A
MinimOS	Ja	Ja	N/A
OpenSUSE	N/A	Ja	N/A
Oracle Linux	N/A	Ja	N/A
Photon OS	N/A	Ja	N/A
RHEL	N/A	Ja	Ja
Rocky Linux	N/A	Ja	Ja

Plattform	Binär	Quelle	Streamen
SLES	N/A	Ja	N/A
Ubuntu	Ja	Ja	N/A
Windows	N/A	N/A	N/A

APK-based Sammlung von OS-Paketen

Dieser Abschnitt enthält die unterstützten Plattformen und die wichtigsten Funktionen für die Sammlung von APK Paketen auf Basis von Betriebssystemen. Weitere Informationen finden Sie unter Alpine Package Keeper auf der Alpine Linux Website.

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt.

- Alpine Linux

> **Note**
>
> Bei APK basierten Systemen sammelt der Amazon Inspector SBOM Generator Paketmetadaten aus der `/lib/apk/db/`Datei.

Schlüssel-Features

- Sammlung von Paketnamen — Extrahiert den Namen jedes installierten Pakets
- Versionssammlung — Extrahiert die Version jedes installierten Pakets
- Identifizierung des Quellpakets — Identifiziert das Quellpaket für jedes installierte Paket

Beispiel

Der folgende Ausschnitt ist ein Beispiel für eine APK Datenbankdatei.

```
C:Q1JlboSJkrN4qkDcokr4zenpcWEXQ=
P:zlib
V:1.2.13-r1
A:x86_64
S:54253
I:110592
T:A compression/decompression Library
U:https://zlib.net/
L:Zlib
o:zlib
```

DPKG-based Sammlung von Betriebssystempaketen

Dieser Abschnitt enthält die unterstützten Plattformen und die wichtigsten Funktionen für die Sammlung von DPKG Paketen auf Basis von Betriebssystemen. Weitere Informationen finden Sie im Debian-Paket auf der Debian Website.

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt.

- Debian
- Ubuntu

> **Note**
>
> Bei DPKG basierten Systemen sammelt der Amazon Inspector SBOM Generator Paketmetadaten aus der `/var/lib/dpkg/status`Datei.

Schlüssel-Features

Im Folgenden sind die wichtigsten Funktionen für DPKG basierte Betriebssystempakete aufgeführt.

- Sammlung von Paketnamen — Extrahiert den Namen jedes installierten Pakets
- Versionssammlung — Extrahiert die Version jedes installierten Pakets
- Identifizierung des Quellpakets — Identifiziert das Quellpaket für jedes installierte Paket

Beispiel

Der folgende Ausschnitt ist ein Beispiel für eine Datei. `/var/lib/dpkg/`

```
Package: zlib1g
Status: install ok installed
Priority: optional
Section: libs
Installed-Size: 168
Maintainer: Mark Brown <broonie@debian.org>
Architecture: amd64
Multi-Arch: same
Source: zlib
Version: 1:1.2.13.dfsg-1
Provides: libz1
Depends: libc6 (>= 2.14)
Breaks: libxml2 (<< 2.7.6.dfsg-2), texlive-binaries (<< 2009-12)
Conflicts: zlib1 (<= 1:1.0.4-7)
Description: compression library - runtime
 zlib is a library implementing the deflate compression method found
 in gzip and PKZIP.  This package includes the shared library.
Homepage: http://zlib.net/
```

RPM-based Sammlung von Betriebssystempaketen

Dieser Abschnitt enthält die unterstützten Plattformen und die wichtigsten Funktionen für die Sammlung von RPM Paketen auf Basis von Betriebssystemen. Weitere Informationen finden Sie unter RPM Package Manager auf der RPM Website.

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt.

- Alma Linux
- Amazon Linux
- CentOS
- Fedora
- OpenSUSE

- Oracle Linux
- PhotonOS
- RedHat Enterprise Linux
- Rocky Linux
- SUSE Linux Enterprise Server

> **Note**
>
> Bei RPM basierten Systemen sammelt der Amazon Inspector SBOM Generator Paketmetadaten aus der `/var/lib/rpm`Datei.

Schlüssel-Features

Im Folgenden sind die wichtigsten Funktionen für Paketsammlungen RPM auf Basis von Betriebssystemen aufgeführt.

- Sammlung von Paketnamen — Extrahiert den Namen jedes installierten Pakets
- Versionssammlung — Extrahiert die Version jedes installierten Pakets
- Identifizierung des Quellpakets — Identifiziert das Quellpaket für jedes installierte Paket
- Stream-Unterstützung — Extrahiert Stream-Metadaten jedes installierten Pakets

Beispiel

Im Folgenden finden Sie ein Beispiel für einen RPM Datenbankdateiausschnitt.

```
/usr/lib/sysimage/rpm/rpmdb.sqlite
/usr/lib/sysimage/rpm/Packages
/usr/lib/sysimage/rpm/Packages.db
/var/lib/rpm/rpmdb.sqlite
/var/lib/rpm/Packages
/var/lib/rpm/Packages.db
```

Sammlung der Windows-Betriebssystemversionen

Im Gegensatz zu Linux-based Betriebssystemen verwendet Windows kein Paketverwaltungssystem für das Betriebssystem selbst. Der Amazon Inspector SBOM Generator sammelt nur die Versionsinformationen des Windows-Betriebssystems. Verwenden Sie für das Scannen von Windows-Anwendungen stattdessen den windows-apps Scanner. Der windows-apps Scanner sammelt Informationen über installierte Anwendungen auf Windows-Systemen. Weitere Informationen finden Sie unterMicrosoft-Anwendungen Sammlung von Ökosystemen.

Schlüssel-Features

- Erfassung der Betriebssystemversionen — Extrahiert die Windows-Betriebssystemversion aus der Windows-Registrierung. Die extrahierte Betriebssystemversion wird zur Erkennung von Sicherheitslücken für Windows-Betriebssysteme verwendet.

Registrierungsschlüssel und -werte

Die folgenden Windows-Registrierungsschlüssel und -werte werden verwendet, um Informationen über den Namen und die Version des Betriebssystems zu sammeln.

- Registrierungsschlüssel

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
```

- Registrierungswert
  - ProductName— Name und Edition des Betriebssystems (z. B."Windows Server 2025 Datacenter")
  - CurrentMajorVersionNumber— die Hauptversion von OS
  - CurrentMinorVersionNumber— Die Nebenversion von OS
  - CurrentBuild— Die Build-Nummer des Betriebssystems
  - UBR— Die Revisionsnummer des Betriebssystems

Sammlung von Chainguard-Image-Paketen

Dieser Abschnitt enthält die unterstützten Plattformen und wichtigsten Funktionen für die Sammlung von Chainguard Image-Paketen. Weitere Informationen finden Sie unter Bilder auf der Chainguard Website.

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt

- Wolfi Linux

> **Note**
>
> Für Chainguard Bilder sammelt der Amazon Inspector SBOM Generator Paketmetadaten aus der `/lib/apk/db/installed` Datei.

Schlüssel-Features

Im Folgenden sind die wichtigsten Funktionen aufgeführt.

- Sammlung von Paketnamen — Extrahiert den Namen jedes installierten Pakets
- Versionssammlung — Extrahiert die Version jedes installierten Pakets
- Identifizierung des Quellpakets — Identifiziert das Quellpaket für jedes installierte Paket

Beispiel

Der folgende Ausschnitt ist ein Beispiel für eine Chainguard Bilddatei.

```
P:wolfi-keys
V:1-r8
A:x86_64
L:MIT
T:Wolfi signing keyring
o:wolfi-keys
```

Sammlung von Image-Paketen ohne Distribution

DistrolessContainer sind Container-Images, die Paketmanager, Shells und andere Hilfsprogramme in Linux Distributionen ausschließen. DistrolessContainer enthalten nur wichtige Abhängigkeiten, die für die Ausführung der Anwendung und die Verbesserung von Leistung und Sicherheit erforderlich sind.

> **Note**
>
> Für DistrolessBilder sammelt der Amazon Inspector SBOM Generator Paketmetadaten aus der `/var/lib/dpkg/status.d` Datei. Es werden nur Distributionen Debian unterstützt, die auf dem Server Ubuntu basieren. Diese können anhand des `NAME` Felds im `/etc/os-release` Dateisystem identifiziert werden, das "" oder Debian "“ anzeigtUbuntu.

Schlüssel-Features

- Sammlung von Paketnamen — Extrahiert den Namen jedes installierten Pakets
- Versionssammlung — Extrahiert die Version jedes installierten Pakets

Beispiel

Das Folgende ist ein Beispiel für eine Distroless Image-Datei.

```
Package: tzdata
Version: 2021a-1+deb11u10
Architecture: all
Maintainer: GNU Libc Maintainers <debian-glibc@lists.debian.org>
Installed-Size: 3413
Depends: debconf (>= 0.5) | debconf-2.0
Provides: tzdata-bullseye
Section: localization
Priority: required
Multi-Arch: foreign
Homepage: https://www.iana.org/time-zones
Description: time zone and daylight-saving time data
 This package contains data required for the implementation of
 standard local time for many representative locations around the
 globe. It is updated periodically to reflect changes made by
 political bodies to time zone boundaries, UTC offsets, and
 daylight-saving rules.
```

MiniMOS-Paketsammlung

Dieser Abschnitt enthält die unterstützten Plattformen und wichtigsten Funktionen für die Sammlung von Minimus Image-Paketen. Weitere Informationen finden Sie auf der Minimus-Website.

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt.

- MinimOS

> **Note**
>
> Für Minimus Bilder sammelt der Amazon Inspector SBOM Generator Paketmetadaten aus der `/lib/apk/db/installed` Datei.

Schlüssel-Features

Im Folgenden sind die wichtigsten Funktionen aufgeführt.

- Sammlung von Paketnamen — Extrahiert den Namen jedes installierten Pakets
- Versionssammlung — Extrahiert den Namen jedes installierten Pakets
- Identifizierung des Quellpakets — Identifiziert das Quellpaket für jedes installierte Paket

Das Folgende ist ein Ausschnitt aus einer Minimus Bilddatei.

```
P:ca-certificates-bundle
V:20241121-r1
A:aarch64
L:MPL-2.0 AND MIT
T:
o:ca-certificates
```

Gehärtete Docker-Images Sammlung von Paketen

Docker Hardened Images(DHI) sind minimale, sicherheitsverstärkte Container-Images. Der Amazon Inspector SBOM Generator erkennt DHI Bilder und Namespaces, unter denen ihre Pakete stehen, `dhi` sodass Sicherheitslücken mit der gehärteten Distribution abgeglichen werden.

Erkennung

DHIBilder der Version 2 werden `ID=dhi` in der `/etc/os-release` Datei angezeigt, wobei das zugrunde liegende Basisbetriebssystem angegeben ist `ID_LIKE=alpine` oder `ID_LIKE=debian` angegeben wird. Der Amazon Inspector SBOM Generator erkennt DHI Bilder direkt anhand dieses `ID` Werts und analysiert das Feld nicht. `PRETTY_NAME`

Beispiel **`etc/os/-Veröffentlichung`** file

Das Folgende ist ein Beispiel für eine `/etc/os-release` Datei für ein Debian basiertes DHI Image.

```
ID=dhi
ID_LIKE=debian
```

Paket-Namespace

Jedes Package in einem DHI Bild hat einen Namensraum unter`dhi`, was eine Paket-URL der Form oder ergibt. `pkg:deb/dhi/<name>@<version>` `pkg:apk/dhi/<name>@<version>` Der PURL-Typ (`deb`oder`apk`) spiegelt das zugrunde liegende Basisbetriebssystem wider.

Beispiel für eine PURL

Im Folgenden finden Sie Beispielpaket-URLs für Pakete in DHI Bildern.

```
# Debian-based DHI image
pkg:deb/dhi/<name>@<version>

# Alpine-based DHI image
pkg:apk/dhi/<name>@<version>
```

Hardened-image Markierungen

Wenn der Amazon Inspector SBOM Generator erkennt`ID=dhi`, kommentiert er das Bild mit gehärteten Bildmarkierungen:

- Der Name der Betriebssystemkomponente ist mit einer Anmerkung versehen — zum Beispiel. `<base> (Docker Hardened Image)` `Debian GNU/Linux (Docker Hardened Image)`
- Der Container oder die Betriebssystemkomponente trägt die Eigenschaft mit dem Wert. `amazon:inspector:sbom_generator:hardened_image:vendor` `Docker`

> **Note**
>
> DHIDie Erkennung gilt für Container-, Localhost- und Volume-Scans.

Gehärtete Red Hat-Images Sammlung von Paketen

Red Hat Hardened Imagessind minimale, sicherheitsverstärkte Container-Images, auf denen sie aufbauen. Hummingbird OS Der Amazon Inspector SBOM Generator erkennt diese Images und die Namespaces, unter denen ihre Pakete stehen, `hummingbird` sodass Sicherheitslücken mit der gehärteten Distribution abgeglichen werden.

Erkennung

Red Hat Hardened ImagesBericht in der Datei. `ID=hummingbird` `/etc/os-release` Der Amazon Inspector SBOM Generator erkennt diese Bilder direkt anhand dieses `ID` Werts.

Beispiel **`/-Veröffentlichung etc/os`** file

Das Folgende ist ein Beispiel für eine `/etc/os-release` Datei für a. Red Hat Hardened Image

```
ID=hummingbird
```

Paket-Namespace

Jedes Package in a Red Hat Hardened Image hat einen Namensraum unter`hummingbird`, was eine Paket-URL der Form ergibt. `pkg:rpm/hummingbird/<name>@<version>`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein Paket in einemRed Hat Hardened Image.

```
pkg:rpm/hummingbird/<name>@<version>
```

Hardened-image Markierungen

Wenn der Amazon Inspector SBOM Generator erkennt`ID=hummingbird`, kommentiert er das Bild mit gehärteten Bildmarkierungen:

- Der Container oder die Betriebssystemkomponente trägt die Eigenschaft mit dem Wert. `amazon:inspector:sbom_generator:hardened_image:vendor` `Red Hat`

> **Note**
>
> Red Hat Hardened ImagesDie Erkennung gilt für Container-, Localhost- und Volume-Scans.

CleanStart Sammlung von Paketen mit gehärteten Bildern

CleanStartImages sind minimale, sicherheitsverstärkte Container-Images, die auf einer Alpine Basis basieren. Der Amazon Inspector SBOM Generator erkennt diese Images und die Namespaces, unter denen ihre Pakete stehen, `clnstrt` sodass Sicherheitslücken mit der gehärteten Distribution abgeglichen werden.

Erkennung

CleanStartBilder werden in der Datei gemeldet. `ID=clnstrt` `/etc/os-release` Der Amazon Inspector SBOM Generator erkennt diese Bilder direkt anhand dieses `ID` Werts.

Beispiel **`/-Veröffentlichung etc/os`** file

Das Folgende ist ein Beispiel für eine `/etc/os-release` Datei für ein CleanStart Bild.

```
ID=clnstrt
ID_LIKE=alpine
```

Paket-Namespace

Jedes Package in einem CleanStart Bild hat einen Namensraum unter`clnstrt`, was eine Paket-URL der Form ergibt. `pkg:apk/clnstrt/<name>@<version>`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein Paket in einem CleanStart Bild.

```
pkg:apk/clnstrt/<name>@<version>
```

Hardened-image Markierungen

Wenn der Amazon Inspector SBOM Generator erkenntID=clnstrt, kommentiert er das Bild mit gehärteten Bildmarkierungen:

- Der Container oder die Betriebssystemkomponente trägt die Eigenschaft mit dem Wert. amazon:inspector:sbom_generator:hardened_image:vendor CleanStart

> **Note**
>
> CleanStartDie Erkennung gilt für Container-, Localhost- und Volume-Scans.

Sammlung von Abhängigkeiten zu Programmiersprachen

Der Amazon Inspector SBOM Generator unterstützt verschiedene Programmiersprachen und Frameworks, die eine robuste und detaillierte Sammlung von Abhängigkeiten bilden. Die Generierung einer SBOM hilft Ihnen dabei, die Zusammensetzung Ihrer Software zu verstehen, sodass Sie Schwachstellen identifizieren und die Einhaltung der Sicherheitsstandards sicherstellen können. Der Amazon Inspector SBOM Generator unterstützt die folgenden Programmiersprachen und Dateiformate.

Gehen Sie zum Scannen von Abhängigkeiten

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
Go	Go	go.mod	N/A	N/A	N/A	N/A	Ja
		go.sum	N/A	N/A	N/A	N/A	Ja
		Go Binaries	Ja	N/A	N/A	N/A	Ja
			N/A	N/A	N/A	N/A	Nein

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
		GOMODCACHE					

gehen. mod/go. Summe

Verwenden Sie `go.sum` Dateien `go.mod` und, um Abhängigkeiten in Go Projekten zu definieren und zu sperren. Der Amazon Inspector SBOM Generator verwaltet diese Dateien je nach Version der Go Toolchain unterschiedlich.

Schlüssel-Features

- Sammelt Abhängigkeiten von `go.mod` (wenn die Go Toolketten-Version 1.17 oder höher ist)
- Sammelt Abhängigkeiten von `go.sum` (wenn die Go Toolketten-Version 1.17 oder niedriger ist)
- Analysiert`go.mod`, um alle deklarierten Abhängigkeiten und Abhängigkeitsversionen zu identifizieren

Beispiel **`go.mod`** file

Das Folgende ist ein Beispiel für eine Datei. `go.mod`

```
module example.com/project

go 1.17

require (
github.com/gin-gonic/gin v1.7.2
golang.org/x/crypto v0.0.0-20210616213533-5cf6c0f8e123
)
```

Beispiel **`go.sum`** file

Das Folgende ist ein Beispiel für eine Datei. `go.sum`

```
github.com/gin-gonic/gin v1.7.2 h1:VZ7DdRl0sghbA6lVGSkX+UXO2+J0aH7RbsNugG+FA8Q=
github.com/gin-gonic/gin v1.7.2/go.mod h1:ILZ1Ngh2f1pL1ASUj7gGk8lGFeNC8cRTaN2ZhsBNbXU=
golang.org/x/crypto v0.0.0-20210616213533-5cf6c0f8e123 h1:b6rCu+qHze
+BUsmC3CZzH8aNu8LzPZTVsNTo64OypSc=
golang.org/x/crypto v0.0.0-20210616213533-5cf6c0f8e123/go.mod h1:K5Dkpb0Q4ewZW/
EzWlQphgJcUMBCzoWrLfDOVzpTGVQ=
```

> **Note**
>
> Jede dieser Dateien erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Gehe zu Binärdateien

Der Amazon Inspector SBOM Generator extrahiert Abhängigkeiten aus kompilierten Go Binärdateien, um sicherzustellen, dass der verwendete Code verwendet wird.

> **Note**
>
> Der Amazon Inspector SBOM Generator unterstützt das Erfassen und Auswerten von Toolchainversionen aus Go Binärdateien, die mit dem offiziellen Compiler erstellt wurden. Go Weitere Informationen finden Sie auf der Website unter Herunterladen und Installieren. Go Wenn Sie beispielsweise die Go Toolchain eines anderen Anbieters verwenden, ist die Bewertung aufgrund möglicher Unterschiede bei der Verteilung und der Verfügbarkeit von Metadaten möglicherweise nicht korrekt. Red Hat

Schlüssel-Features

- Extrahiert Abhängigkeitsinformationen direkt aus Binärdateien Go
- Sammelt in der Binärdatei eingebettete Abhängigkeiten

- Erkennt und extrahiert die Version der Go Toolchain, die zum Kompilieren der Binärdatei verwendet wurde.

GOMODCACHE

Der Amazon Inspector SBOM Generator scannt den Go Modul-Cache, um Informationen über installierte Abhängigkeiten zu sammeln. In diesem Cache werden heruntergeladene Module gespeichert, um sicherzustellen, dass dieselben Versionen in verschiedenen Builds verwendet werden.

Schlüssel-Features

- Durchsucht das `GOMODCACHE` Verzeichnis, um zwischengespeicherte Module zu identifizieren
- Extrahiert detaillierte Metadaten, einschließlich Modulnamen, Versionen und Quell-URLs

Beispiel für eine Struktur

Das Folgende ist ein Beispiel für die `GOMODCACHE` Struktur.

```
~/go/pkg/mod/
### github.com/gin-gonic/gin@v1.7.2
### golang.org/x/crypto@v0.0.0-20210616213533-5cf6c0f8e123
```

> **Note**
>
> Diese Struktur erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben, und sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Scannen von Java-Abhängigkeiten

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolkette-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
`Java`	`Maven`	Kompilierte Java Anwendungen (.jar/.war/.ear) `pom.xml`	N/A N/A	N/A N/A	Ja Ja	N/A N/A	Ja Ja

> **Note**
>
> Unsere Funktion zur Bewertung von Sicherheitslücken unterstützt nur das Maven Central-Repository. Third-party Repositorys wieJBoss Enterprise Maven Repository, werden derzeit nicht unterstützt.

Der Amazon Inspector SBOM Generator führt Java Abhängigkeitsscans durch, indem er kompilierte Java Anwendungen und `pom.xml` Dateien analysiert. Beim Scannen kompilierter Anwendungen generiert der Scanner SHA-1-Hashes zur Integritätsprüfung, extrahiert eingebettete `pom.properties` Dateien und analysiert verschachtelte Dateien. `pom.xml`

SHA-1-Hash-Sammlung (für kompilierte .jar-, .WAR-, .ear-Dateien)

Der Amazon Inspector SBOM Generator versucht, SHA-1-Hashes für alle `.ear`,, und `.war` Dateien in einem Projekt zu sammeln`.jar`, um die Integrität und Rückverfolgbarkeit der kompilierten Artefakte zu gewährleisten. Java

Schlüssel-Features

- Generiert SHA—1-Hashes für alle kompilierten Artefakte Java

Beispiel für ein Artefakt

Im Folgenden finden Sie ein Beispiel für ein SHA-1-Artefakt.

```
{
  "bom-ref": "comp-52",
  "type": "library",
  "name": "jul-to-slf4j",
  "version": "2.0.6",
  "hashes": [
    {
      "alg": "SHA-1",
      "content": ""
    }
  ],
  "purl": "pkg:maven/jul-to-slf4j@2.0.6",
  "properties": [
    {
      "name": "amazon:inspector:sbom_generator:source_path",
      "value": "test-0.0.1-SNAPSHOT.jar/BOOT-INF/lib/jul-to-slf4j-2.0.6.jar"
    }
  ]
}
```

> **Note**
>
> Dieses Artefakt erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben, und sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

pom.properties

Die `pom.properties` Datei wird in Maven Projekten verwendet, um Projektmetadaten, einschließlich Paketnamen und Paketversionen, zu speichern. Der Amazon Inspector SBOM Generator analysiert diese Datei, um Projektinformationen zu sammeln.

Schlüssel-Features

- Analysiert und extrahiert Paketartefakte, Paketgruppen und Paketversionen

Beispiel **`pom.properties`** file

Im Folgenden wird ein Beispiel für eine `pom.properties`-Datei dargestellt.

```
#Generated by Maven
#Tue Mar 16 15:44:02 UTC 2021

version=1.6.0
groupId=net.datafaker
artifactId=datafaker
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Ohne verschachtelte **`pom.xml`** Parsing

Wenn Sie das `pom.xml` Parsen beim Scannen kompilierter Java Anwendungen ausschließen möchten, verwenden Sie das `--skip-nested-pomxml` Argument.

pom.xml

Die `pom.xml` Datei ist die zentrale Konfigurationsdatei für Maven Projekte. Sie enthält Informationen über Projekte und Projektabhängigkeiten. Der Amazon Inspector SBOM Generator analysiert pom.xml Dateien, um Abhängigkeiten zu erfassen. Dabei werden eigenständige Dateien in Repositorys und Dateien in kompilierten Dateien gescannt. .jar

Schlüssel-Features

- Analysiert und extrahiert Paketartefakte, Paketgruppen und Paketversionen aus Dateien. `pom.xml`

Unterstützt Maven Bereiche und Tags

Abhängigkeiten werden in den folgenden Maven Bereichen erfasst:

- compile
- bereitgestellt
- runtime
- Test
- system
- Import

Abhängigkeiten werden mit dem folgenden Maven Tag erfasst:`<optional>true</optional>`.

Beispiel **`pom.xml`** Datei mit einem Gültigkeitsbereich

Das Folgende ist ein Beispiel für eine `pom.xml` Datei mit einem Bereich.

```
<dependency>
<groupId>jakarta.servlet</groupId>
<artifactId>jakarta.servlet-api</artifactId>
</version>6.0.0</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.28</version>
<scope>runtime</scope>
</dependency>
```

Beispiel **`pom.xml`** Datei ohne Gültigkeitsbereich

Im Folgenden finden Sie ein Beispiel für eine `pom.xml` Datei ohne Bereich.

```
<dependency>
<groupId>com.fasterxml.jackson.core</groupId>
```

```
<artifactId>jackson-databind</artifactId>
<version>2.17.1</version>
</dependency>

<dependency>
<groupId>org.jenkins-ci.plugins</groupId>
<artifactId>plain-credentials</artifactId>
<version>183.va_de8f1dd5a_2b_</version>
</dependency>

<dependency>
<groupId>org.jenkins-ci.plugins</groupId>
<artifactId>jackson2-api</artifactId>
<version>2.15.2-350.v0c2f3f8fc595</version>
</dependency>
```

> **Note**
>
> Jede dieser Dateien erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

JavaScript Scannen von Abhängigkeiten

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolkette-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
`Javascript`	`Node Modules`	`node_modules/*/package.json`	N/A	N/A	Ja	Ja	Ja
	`NPM`		N/A	Ja	N/A	N/A	Nein
	`PNPM`		N/A	Ja	N/A	N/A	Nein
			N/A	Ja	N/A	N/A	Nein

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv

	YARN	`package-lock.json` (v1, v2, and v3) / `npm-shrinkwrap.json`
		
		`pnpm-lock.yaml`
		
		`yarn.lock`					

package.json

Die `package.json` Datei ist ein Kernbestandteil von Node.js Projekten. Sie enthält Metadaten zu installierten Paketen. Der Amazon Inspector SBOM Generator scannt diese Datei, um Paketnamen und Paketversionen zu identifizieren.

Schlüssel-Features

- Analysiert die JSON-Dateistruktur, um Paketnamen und Versionen zu extrahieren
- Identifiziert private Pakete mit privaten Werten

Beispiel **`package.json`** file

Im Folgenden wird ein Beispiel für eine `package.json`-Datei dargestellt.

```
{
"name": "arrify",
"private": true,
"version": "2.0.1",
"description": "Convert a value to an array",
"license": "MIT",
"repository": "sindresorhus/arrify"
}
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

package-lock.json

Die `package-lock.json` Datei wird automatisch von npm generiert, um genaue Versionen der für ein Projekt installierten Abhängigkeiten zu sperren. Es gewährleistet die Konsistenz in Umgebungen, indem exakte Versionen aller Abhängigkeiten und ihrer Unterabhängigkeiten gespeichert werden. Diese Datei kann zwischen regulären Abhängigkeiten und Entwicklungsabhängigkeiten unterscheiden.

Schlüssel-Features

- Analysiert die JSON-Dateistruktur, um Paketnamen und Paketversionen zu extrahieren
- Unterstützt die Erkennung von Abhängigkeiten von Entwicklern

Beispiel **`package-lock.json`** file

Im Folgenden wird ein Beispiel für eine `package-lock.json`-Datei dargestellt.

```
"verror": {
"version": "1.10.0",
```

```
"resolved": "https://registry.npmjs.org/verror/-/verror-1.10.0.tgz",
"integrity": "sha1-OhBcoXBTr1XW4nDB+CiGguGNpAA=",
"requires": {
    "assert-plus": "^1.0.0",
    "core-util-is": "1.0.2",
    "extsprintf": "^1.2.0"
}
},
"wrappy": {
"version": "1.0.2",
"resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
"integrity": "sha1-tSQ9jz7BqjXxNkYFvA0QNuMKtp8=",
"dev": true
},
"yallist": {
"version": "3.0.2",
"resolved": "https://registry.npmjs.org/yallist/-/yallist-3.0.2.tgz",
"integrity": "sha1-hFK0u36Dx8GI2AQcGoN8dz1ti7k="
}
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

npm-shrinkwrap.json

npmgeneriert `package-lock.json` automatisch `npm-shrinkwrap.json` Dateien, um exakte Versionen der für ein Projekt installierten Abhängigkeiten zu sperren. Dies garantiert Konsistenz in Umgebungen, indem exakte Versionen aller Abhängigkeiten und Unterabhängigkeiten gespeichert werden. Die Dateien unterscheiden zwischen regulären Abhängigkeiten und Entwicklungsabhängigkeiten.

Schlüssel-Features

- Analysieren Sie die `package-lock` Versionen 1, 2 und 3 der JSON Dateistruktur, um den Paketnamen und die Version zu extrahieren

- Die Erkennung von Abhängigkeiten zwischen Entwicklern wird unterstützt (`package-lock.json`erfasst Produktions- und Entwicklungsabhängigkeiten, sodass Tools erkennen können, welche Pakete in Entwicklungsumgebungen verwendet werden)
- Die `npm-shrinkwrap.json` Datei hat Vorrang vor der `package-lock.json` Datei

Beispiel

Im Folgenden wird ein Beispiel für eine `package-lock.json`-Datei dargestellt.

```
"verror": {
            "version": "1.10.0",
            "resolved": "https://registry.npmjs.org/verror/-/verror-1.10.0.tgz",
            "integrity": "sha1-OhBcoXBTr1XW4nDB+CiGguGNpAA=",
            "requires": {
                "assert-plus": "^1.0.0",
                "core-util-is": "1.0.2",
                "extsprintf": "^1.2.0"
            }
        },
        "wrappy": {
            "version": "1.0.2",
            "resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
            "integrity": "sha1-tSQ9jz7BqjXxNkYFvA0QNuMKtp8=",
            "dev": true
        },
        "yallist": {
            "version": "3.0.2",
            "resolved": "https://registry.npmjs.org/yallist/-/yallist-3.0.2.tgz",
            "integrity": "sha1-hFK0u36Dx8GI2AQcGoN8dz1ti7k="
}
```

pnpm-yaml.lock

Die `pnpm-lock.yaml` Datei wird von pnpm generiert, um die installierten Abhängigkeitsversionen aufzuzeichnen. Außerdem werden Entwicklungsabhängigkeiten separat verfolgt.

Schlüssel-Features

- Analysiert die YAML-Dateistruktur, um Paketnamen und Versionen zu extrahieren
- Unterstützt die Erkennung von Abhängigkeiten von Entwicklern

Beispiel

Im Folgenden wird ein Beispiel für eine `pnpm-lock.yaml`-Datei dargestellt.

```
lockfileVersion: 5.3
importers:
my-project:
dependencies:
  lodash: 4.17.21
devDependencies:
  jest: 26.6.3
specifiers:
  lodash: ^4.17.21
  jest: ^26.6.3
packages:
/lodash/4.17.21:
resolution:
  integrity: sha512-xyz
engines:
  node: '>=6'
dev: false
/jest/26.6.3:
resolution:
  integrity: sha512-xyz
dev: true
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

yarn.lock

Der Amazon Inspector SBOM Generator versucht, SHA-1-Hashes für `.ear`, und `.war` Dateien in einem Projekt zu sammeln`.jar`, um die Integrität und Rückverfolgbarkeit kompilierter Artefakte zu gewährleisten. Java

Schlüssel-Features

- Generiert SHA—1-Hashes für alle kompilierten Artefakte Java

Beispiel für ein SHA-1-Artefakt

Im Folgenden finden Sie ein Beispiel für ein SHA-1-Artefakt.

```
"@ampproject/remapping@npm:^2.2.0":
version: 2.2.0
resolution: "@ampproject/remapping@npm:2.2.0"
dependencies:
"@jridgewell/gen-mapping": ^0.1.0
"@jridgewell/trace-mapping": ^0.3.9
checksum:
 d74d170d06468913921d72430259424b7e4c826b5a7d39ff839a29d547efb97dc577caa8ba3fb5cf023624e9af9d09
languageName: node
linkType: hard

"@babel/code-frame@npm:^7.0.0, @babel/code-frame@npm:^7.12.13, @babel/code-
frame@npm:^7.18.6, @babel/code-frame@npm:^7.21.4":
version: 7.21.4
resolution: "@babel/code-frame@npm:7.21.4"
dependencies:
"@babel/highlight": ^7.18.6
checksum:
 e5390e6ec1ac58dcef01d4f18eaf1fd2f1325528661ff6d4a5de8979588b9f5a8e852a54a91b923846f7a5c681b217
languageName: node
linkType: hard
```

> **Note**
>
> Dieses Artefakt erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben, und sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

.NET-Abhängigkeiten scannen

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
.NET	.NET Core	*.deps.json	N/A	N/A	N/A	N/A	Ja
	Nuget	Packages.config	N/A	N/A	N/A	N/A	Ja
	Nuget	packages.lock.json	N/A	N/A	Ja	N/A	Ja
	.NET	.csproj	N/A	N/A	N/A	N/A	Ja

Packages.config

Die `Packages.config` Datei ist eine XML-Datei, die von einer älteren Version von Nuget zur Verwaltung von Projektabhängigkeiten verwendet wird. Sie listet alle Pakete auf, auf die das Projekt verweist, einschließlich bestimmter Versionen.

Schlüssel-Features

- Analysiert die XML-Struktur, um Paket-IDs und Versionen zu extrahieren

Beispiel

Im Folgenden wird ein Beispiel für eine `Packages.config`-Datei dargestellt.

```
<?xml version="1.0" encoding="utf-8"? >
<packages>
<package id="FluentAssertions" version="5.4.1" targetFramework="net461" />
```

```
<package id="Newtonsoft.Json" version="11.0.2" targetFramework="net461" />
<package id="SpecFlow" version="2.4.0" targetFramework="net461" />
<package id="SpecRun.Runner" version="1.8.0" targetFramework="net461" />
<package id="SpecRun.SpecFlow" version="1.8.0" targetFramework="net461" />
<package id="SpecRun.SpecFlow.2-4-0" version="1.8.0" targetFramework="net461" />
<package id="System.ValueTuple" version="4.5.0" targetFramework="net461" />
</packages>
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

*.deps.json

Die `*.deps.json` Datei wird von .NET Core Projekten generiert und enthält detaillierte Informationen zu allen Abhängigkeiten, einschließlich Pfaden, Versionen und Laufzeitabhängigkeiten. Diese Datei stellt sicher, dass die Runtime über die notwendigen Informationen verfügt, um die richtigen Versionen von Abhängigkeiten zu laden.

Schlüssel-Features

- Analysiert die JSON-Struktur nach umfassenden Abhängigkeitsdetails
- Extrahiert Paketnamen und Versionen in eine `libraries` Liste.

Beispiel **`.deps.json`** file

Im Folgenden wird ein Beispiel für eine `.deps.json`-Datei dargestellt.

```
{
"runtimeTarget": {
    "name": ".NETCoreApp,Version=v7.0",
    "signature": ""
},
```

```
"libraries": {
    "sample-Nuget/1.0.0": {
        "type": "project",
        "serviceable": false,
        "sha512": ""
    },
    "Microsoft.EntityFrameworkCore/7.0.5": {
        "type": "package",
        "serviceable": true,
        "sha512": "sha512-
RXbRLHHWP2Z3pq8qcL5nQ6LPeoOyp8hasM5bd0Te8PiQi3RjWQR4tcbdY5XMqQ+oTO9wA8/RLhZRn/
hnxlTDnQ==",
        "path": "microsoft.entityframeworkcore/7.0.5",
        "hashPath": "microsoft.entityframeworkcore.7.0.5.nupkg.sha512"
    },
}
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

packages.lock.json

Die `packages.lock.json` Datei wird von neueren Versionen von verwendetNuget, um genaue Versionen von Abhängigkeiten für ein .NET Projekt zu sperren, um sicherzustellen, dass dieselben Versionen in verschiedenen Umgebungen konsistent verwendet werden.

Schlüssel-Features

- Analysiert die JSON-Struktur, um gesperrte Abhängigkeiten aufzulisten
- Unterstützt sowohl direkte als auch transitive Abhängigkeiten
- Extrahiert den Paketnamen und die aufgelösten Versionen

Beispiel **`packages.lock.json`** file

Im Folgenden wird ein Beispiel für eine `packages.lock.json`-Datei dargestellt.

```
{
"version": 1,
"dependencies": {
"net7.0": {
  "Microsoft.EntityFrameworkCore": {
    "type": "Direct",
    "requested": "[7.0.5, )",
    "resolved": "7.0.5",
    "contentHash": "RXbRLHHWP2Z3pq8qcL5nQ6LPeoOyp8hasM5bd0Te8PiQi3RjWQR4tcbdY5XMqQ
+oTO9wA8/RLhZRn/hnxlTDnQ==",
    "dependencies": {
      "Microsoft.EntityFrameworkCore.Abstractions": "7.0.5",
      "Microsoft.EntityFrameworkCore.Analyzers": "7.0.5",
      "Microsoft.Extensions.Caching.Memory": "7.0.0",
      "Microsoft.Extensions.DependencyInjection": "7.0.0",
      "Microsoft.Extensions.Logging": "7.0.0"
    }
  },
  "Newtonsoft.Json": {
    "type": "Direct",
    "requested": "[13.0.3, )",
    "resolved": "13.0.3",
    "contentHash": "HrC5BXdl00IP9zeV+0Z848QWPAoCr9P3bDEZguI+gkLcBKAOxix/tLEAAHC
+UvDNPv4a2d18lOReHMOagPa+zQ=="
  },
  "Microsoft.Extensions.Primitives": {
    "type": "Transitive",
    "resolved": "7.0.0",
    "contentHash": "um1KU5kxcRp3CNuI8o/GrZtD4AIOXDk
+RLsytjZ9QPok3ttLUelLKpilVPuaFT3TFjOhSibUAso0odbOaCDj3Q=="
  }
}
}
}
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste

anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

.csproj

Die `.csproj` Datei ist in XML geschrieben und die Projektdatei für Projekte. .NET Sie enthält Verweise auf Nuget Pakete, Projekteigenschaften und Build-Konfigurationen.

Schlüssel-Features

- Analysiert die XML-Struktur, um Paketverweise zu extrahieren

Beispiel **`.csproj`** file

Im Folgenden wird ein Beispiel für eine `.csproj`-Datei dargestellt.

```
<Project Sdk="Microsoft.NET.Sdk">
<PropertyGroup>
<TargetFramework>net7.0</TargetFramework>
<RootNamespace>sample_Nuget</RootNamespace>
<ImplicitUsings>enable</ImplicitUsings>
<Nullable>enable</Nullable>
<RestorePackagesWithLockFile>true</RestorePackagesWithLockFile>
</PropertyGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
<PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
<PackageReference Include="Microsoft.EntityFrameworkCore" Version="7.0.5" />
</ItemGroup>
</Project>
```

Beispiel **`.csproj`** file

Im Folgenden wird ein Beispiel für eine `.csproj`-Datei dargestellt.

```
<PackageReference Include="ExamplePackage" Version="6.*" />
<PackageReferencePackageReference Include="ExamplePackage" Version="(4.1.3,)" />
```

```
<PackageReference Include="ExamplePackage" Version="(,5.0)" />
<PackageReference Include="ExamplePackage" Version="[1,3)" />
<PackageReference Include="ExamplePackage" Version="[1.3.2,1.5)" />
```

> **Note**
>
> Jede dieser Dateien erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Scannen von PHP-Abhängigkeiten

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
PHP	Composer	`composer.lock` `/vendor/composer/installed.json`	N/A N/A	N/A N/A	Ja Ja	N/A N/A	Ja Ja

composer.lock

Die `composer.lock` Datei wird automatisch generiert, wenn die Befehle composer install oder composer update ausgeführt werden. Diese Datei garantiert, dass in jeder Umgebung dieselben

Versionen von Abhängigkeiten installiert sind. Dies sorgt für einen konsistenten und zuverlässigen Erstellungsprozess.

Schlüssel-Features

- Analysiert das JSON-Format für strukturierte Daten
- Extrahiert Namen und Versionen von Abhängigkeiten

Beispiel **`composer.lock`** file

Im Folgenden wird ein Beispiel für eine `composer.lock`-Datei dargestellt.

```
{
"packages": [
    {
        "name": "nesbot/carbon",
        "version": "2.53.1",
        // TRUNCATED
    },
    {
        "name": "symfony/deprecation-contracts",
        "version": "v3.2.1",
        // TRUNCATED
    },
    {
        "name": "symfony/polyfill-mbstring",
        "version": "v1.27.0",
        // TRUNCATED
    }
]
// TRUNCATED
}
```

> **Note**
>
> Dadurch wird eine Ausgabe erzeugt, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste

anzugeben, und sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

//installed.json vendor/composer

Die `/vendor/composer/installed.json` Datei befindet sich im `vendor/composer` Verzeichnis und bietet eine umfassende Liste aller installierten Pakete und Paketversionen.

Schlüssel-Features

- Analysiert das JSON-Format für strukturierte Daten
- Extrahiert die Namen und die Version der Abhängigkeiten

Beispiel **`/vendor/composer/installed.json`** file

Im Folgenden wird ein Beispiel für eine `/vendor/composer/installed.json`-Datei dargestellt.

```
{
"packages": [
    {
        "name": "nesbot/carbon",
        "version": "2.53.1",
        // TRUNCATED
    },
    {
        "name": "symfony/deprecation-contracts",
        "version": "v3.2.1",
        // TRUNCATED
    },
    {
        "name": "symfony/polyfill-mbstring",
        "version": "v1.27.0",
        // TRUNCATED
    }
]
// TRUNCATED
}
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Scannen von Python-Abhängigkeiten

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolchain-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
Python	pip	requirements.txt	N/A	N/A	N/A	N/A	Ja
	Poetry	Poetry.lock	N/A	N/A	N/A	N/A	Ja
	Pipenv	Pipfile.lock	N/A	N/A	N/A	N/A	Ja
	uv	uv.lock	N/A	Ja	N/A	N/A	Ja
	Egg/Wheel	.egg-info/PKG-INFO	N/A	N/A	N/A	N/A	Ja
		.dist-info/METADATA	N/A	N/A	N/A	N/A	Ja

requirements.txt

Die `requirements.txt` Datei ist ein in Python Projekten weit verbreitetes Format, um Projektabhängigkeiten zu spezifizieren. Jede Zeile in dieser Datei enthält ein Paket mit seinen Versionseinschränkungen. Der Amazon Inspector SBOM Generator analysiert diese Datei, um Abhängigkeiten genau zu identifizieren und zu katalogisieren.

Schlüssel-Features

- Unterstützt Versionsbezeichner (== und =)
- Unterstützt Kommentare und komplexe Abhängigkeitszeilen

> **Note**
>
> Die Versionsbezeichner <= und => werden nicht unterstützt.

Beispiel **`requirements.txt`** file

Im Folgenden wird ein Beispiel für eine `requirements.txt`-Datei dargestellt.

```
flask==1.1.2
requests==2.24.0
numpy==1.18.5
foo~=1.2.0
# Comment about a dependency
scipy. # invalid
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Pipfile.lock

Pipenvist ein Tool, das das Beste aus allen Verpackungswelten bietet (gebündelt, gepinnt und unverheftet). Es `Pipfile.lock` sperrt genaue Versionen von Abhängigkeiten, um deterministische Builds zu ermöglichen. Der Amazon Inspector SBOM Generator liest diese Datei, um Abhängigkeiten und ihre aufgelösten Versionen aufzulisten.

Schlüssel-Features

- Analysiert das JSON-Format auf die Auflösung von Abhängigkeiten
- Unterstützt Standard- und Entwicklungsabhängigkeiten

Beispiel **`Pipfile.lock`** file

Im Folgenden wird ein Beispiel für eine `Pipfile.lock`-Datei dargestellt.

```
{
"default": {
    "requests": {
        "version": "==2.24.0",
        "hashes": [
            "sha256:cc718bb187e53b8d"
        ]
    }
},
"develop": {
    "blinker": {
        "hashes": [
            "sha256:1779309f71bf239144b9399d06ae925637cf6634cf6bd131104184531bf67c01",
            "sha256:8f77b09d3bf7c795e969e9486f39c2c5e9c39d4ee07424be2bc594ece9642d83"
        ],
        "markers": "python_version >= '3.8'",
        "version": "==1.8.2"
    }
}
}
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Poetry.lock

Poetryist ein Tool zur Verwaltung und Paketierung von Abhängigkeiten für Python. Die `Poetry.lock` Datei sperrt genaue Versionen von Abhängigkeiten, um konsistente Umgebungen zu ermöglichen. Der Amazon Inspector SBOM Generator extrahiert detaillierte Abhängigkeitsinformationen aus dieser Datei.

Schlüssel-Features

- Analysiert das TOML-Format für strukturierte Daten
- Extrahiert Namen und Versionen von Abhängigkeiten

Beispiel **`Poetry.lock`** file

Im Folgenden wird ein Beispiel für eine `Poetry.lock`-Datei dargestellt.

```
[[package]]
name = "flask"
version = "1.1.2"
description = "A simple framework for building complex web applications."
category = "main"
optional = false
python-versions = ">=3.5"
[[package]]
name = "requests"
version = "2.24.0"
description = "Python HTTP for Humans."
category = "main"
optional = false
python-versions = ">=3.5"
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

uv.lock

uvist ein schneller Python Paketmanager, in den geschrieben wurde. Rust Die `uv.lock` Datei sperrt genaue Versionen von Abhängigkeiten, um konsistente Umgebungen zu ermöglichen. Der Amazon Inspector SBOM Generator extrahiert detaillierte Abhängigkeitsinformationen aus dieser Datei.

Schlüssel-Features

- Analysiert das für strukturierte Daten formatierte TOML `uv.lock`
- Extrahiert Namen und Versionen von Abhängigkeiten
- Unterstützt Entwicklungsabhängigkeiten
- Sammelt nur Pakete, deren Quelle eine Registrierung ist

Beispiel **`uv.lock`** file

Im Folgenden wird ein Beispiel für eine `uv.lock`-Datei dargestellt.

```
version = 1
requires-python = ">=3.12"

[[package]]
name = "flask"
version = "3.1.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "blinker" },
    { name = "click" },
    { name = "itsdangerous" },
    { name = "jinja2" },
    { name = "markupsafe" },
```

```
    { name = "werkzeug" },
]

[[package]]
name = "pytest"
version = "8.3.4"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "iniconfig" },
    { name = "packaging" },
    { name = "pluggy" },
]

[package.dev-dependencies]
dev = [
    { name = "pytest" },
]
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Egg/Wheel

Für global installierte Python-Pakete unterstützt der Amazon Inspector SBOM Generator das Parsen von Metadatendateien in den `.egg-info/PKG-INFO` Verzeichnissen und. `.dist-info/METADATA` Diese Dateien enthalten detaillierte Metadaten zu installierten Paketen.

Schlüssel-Features

- Extrahiert den Paketnamen und die Version
- Unterstützt sowohl Eier- als auch Radformate

Erweiterte Standard-Scanpfade (Sbomgen 1.13.0 und höher)

In Amazon Inspector SBOM Generator Version 1.13.0 und höher wurden die standardmäßigen Localhost-Scanpfade des Scanners um `pip` Installationen auf Benutzerebene (`pip install --user`) und Installationen erweitert, die durch den Befehl erzeugt wurden. `uv tool install` Dies ermöglicht die Erkennung von Python-Paketen, die außerhalb des gesamten Systems installiert sind`site-packages`, einschließlich Frameworks und Tools wie CrewAI, Aider, GPT Engineer und Open Interpreter, deren empfohlene Installationsmethoden Installationen an diesen Speicherorten erzeugen (z. B.). `uv tool install crewai`
Standard-Scanpfade wurden hinzugefügt:

- Linux—`/home/*/.local/lib/python*/site-packages/`, `/home/*/.local/share/uv/tools/*/lib/python*/site-packages/` (mit `/root/` Entsprechungen für Container, die als Root ausgeführt werden).
- Windows – `%APPDATA%\Python\Python*\Lib\site-packages\`, `%LOCALAPPDATA%\uv\tools\*\Lib\site-packages\`.

Verwenden Sie für nicht standardmäßige Python Installationspräfixe auf einer beliebigen Plattform das `--path` Argument, um den Scanner auf den Installationsort zu verweisen.

Beispiel **`PKG-INFO/METADATA`** file

Im Folgenden wird ein Beispiel für eine `PKG-INFO/METADATA`-Datei dargestellt.

```
Metadata-Version: 1.2
Name: Flask
Version: 1.1.2
Summary: A simple framework for building complex web applications.
Home-page: https://palletsprojects.com/p/flask/
```

Note

Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste

anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Scannen von Abhängigkeiten in Ruby

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolchaining-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
Ruby	Bundler	Gemfile.lock	N/A	N/A	Ja	N/A	Ja
		.gemspec	N/A	N/A	N/A	N/A	Ja
		globall installed Gems	N/A	N/A	N/A	N/A	Ja

Gemfile.lock

Die Gemfile.lock Datei sperrt genaue Versionen aller Abhängigkeiten, um sicherzustellen, dass in jeder Umgebung dieselben Versionen verwendet werden.

Schlüssel-Features

- Analysiert die Gemfile.lock Datei, um Abhängigkeiten und Abhängigkeitsversionen zu identifizieren
- Extrahiert detaillierte Paketnamen und Paketversionen

Beispiel **Gemfile.lock** file

Im Folgenden wird ein Beispiel für eine Gemfile.lock-Datei dargestellt.

```
GEM
remote: https://rubygems.org/
specs:
ast (2.4.2)
awesome_print (1.9.2)
diff-lcs (1.5.0)
json (2.6.3)
parallel (1.22.1)
parser (3.2.2.0)
nokogiri (1.16.6-aarch64-linux)
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

.gemspec

Die `.gemspec` Datei ist eine RubyGem Datei, die Metadaten zu einem Edelstein enthält. Der Amazon Inspector SBOM Generator analysiert diese Datei, um detaillierte Informationen über einen Edelstein zu sammeln.

Schlüssel-Features

- Analysiert und extrahiert den Edelsteinnamen und die Edelsteinversion

> **Note**
>
> Die Referenzspezifikation wird nicht unterstützt.

Beispiel **`.gemspec`** file

Im Folgenden wird ein Beispiel für eine `.gemspec`-Datei dargestellt.

```
Gem::Specification.new do |s|
s.name        = "generategem"
s.version     = "2.0.0"
s.date        = "2020-06-12"
s.summary     = "generategem"
s.description = "A Gemspec Builder"
s.email       = "edersondeveloper@gmail.com"
s.files       = ["lib/generategem.rb"]
s.homepage    = "https://github.com/edersonferreira/generategem"
s.license     = "MIT"
s.executables = ["generategem"]
s.add_dependency('colorize', '~> 0.8.1')
end
```

```
# Not supported

Gem::Specification.new do |s|
s.name        = &class1
s.version     = &foo.bar.version
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Weltweit installierte Edelsteine

Der Amazon Inspector SBOM Generator unterstützt das Scannen global installierter Gems, die sich in Standardverzeichnissen wie `/usr/local/lib/ruby/gems/<ruby_version>/gems/` Amazon EC2/Amazon ECR und `ruby/gems/<ruby_version>/gems/` Lambda befinden. Dadurch wird sichergestellt, dass alle global installierten Abhängigkeiten identifiziert und katalogisiert werden.

Schlüssel-Features

- Identifiziert und scannt alle global installierten Gems in Standardverzeichnissen

- Extrahiert Metadaten und Versionsinformationen für jedes global installierte Gem

Beispiel für eine Verzeichnisstruktur

Das Folgende ist ein Beispiel für eine Verzeichnisstruktur.

```
.
### /usr/local/lib/ruby/3.5.0/gems/
### actrivesupport-6.1.4
### concurrent-ruby-1.1.9
### i18n-1.8.10
```

> **Note**
>
> Diese Struktur erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben, und sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Scannen von Abhängigkeiten auf Rost

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolchain-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
`Rust`	`Cargo.toml`	`Cargo.toml`	N/A	N/A	N/A	N/A	Ja
		`Cargo.lock`	N/A	N/A	Ja	N/A	Ja
		`Rust binary`	Ja	N/A	N/A	N/A	Ja

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
		(built with cargo-auditable)					

Cargo.toml

Die `Cargo.toml` Datei ist die Manifestdatei für Rust Projekte.

Schlüssel-Features

- Analysiert und extrahiert die `Cargo.toml` Datei, um den Namen und die Version des Projektpakets zu identifizieren.

Beispiel **`Cargo.toml`** file

Im Folgenden wird ein Beispiel für eine `Cargo.toml`-Datei dargestellt.

```
[package]
name = "wait-timeout"
version = "0.2.0"
description = "A crate to wait on a child process with a timeout specified across Unix
 and\nWindows platforms.\n"
homepage = "https://github.com/alexcrichton/wait-timeout"
documentation = "https://docs.rs/wait-timeout"
readme = "README.md"
categories = ["os"]
license = "MIT/Apache-2.0"
repository = "https://github.com/alexcrichton/wait-timeout"
[target."cfg(unix)".dependencies.libc]
version = "0.2"
[badges.appveyor]
```

```
repository = "alexcrichton/wait-timeout"
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Cargo.lock

Die `Cargo.lock` Datei sperrt abhängige Versionen, um sicherzustellen, dass bei der Erstellung eines Projekts dieselben Versionen verwendet werden.

Schlüssel-Features

- Analysiert die `Cargo.lock` Datei, um alle Abhängigkeiten und Abhängigkeitsversionen zu identifizieren.

Beispiel **`Cargo.lock`** file

Im Folgenden wird ein Beispiel für eine `Cargo.lock`-Datei dargestellt.

```
# This file is automatically @generated by Cargo.
# It is not intended for manual editing.
[[package]]
name = "adler32"
version = "1.0.3"
source = "registry+https://github.com/rust-lang/crates.io-index"

[[package]]
name = "aho-corasick"
version = "0.7.4"
source = "registry+https://github.com/rust-lang/crates.io-index"
```

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Rust-Binärdateien mit überprüfbarer Ladung

Der Amazon Inspector SBOM Generator sammelt Abhängigkeiten von Rust Binärdateien, die mit der Bibliothek erstellt wurden. `cargo-auditable` Dies bietet zusätzliche Abhängigkeitsinformationen, indem die Extraktion von Abhängigkeiten aus kompilierten Binärdateien aktiviert wird.

Schlüssel-Features

- Extrahiert Abhängigkeitsinformationen direkt aus Rust Binärdateien, die mit der Bibliothek erstellt wurden `cargo-auditable`
- Ruft Metadaten und Versionsinformationen für Abhängigkeiten ab, die in den Binärdateien enthalten sind

> **Note**
>
> Diese Datei erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Schnelles Scannen von Abhängigkeiten

Programmiersprache	Paketmanager	Unterstützte Artefakte	Toolketten-Unterstützung	Abhängigkeiten bei der Entwicklung	Transitive Abhängigkeiten	Private Flagge	Rekursiv
`Swift`	`Swift Package Manager`	`Package.resolved (v1, v2, and v3)`	N/A	N/A	N/A	N/A	Ja

Package.resolved

Die `Package.resolved` Datei wird vom Swift Package Manager generiert, um genaue Versionen von Abhängigkeiten für ein Swift Projekt zu sperren. Der Amazon Inspector SBOM Generator analysiert diese Datei, um Abhängigkeitsnamen, Versionen und Quell-Repository-Informationen zu sammeln. Die Schemaversionen 1, 2 und 3 des `Package.resolved` Formats werden unterstützt.

> **Note**
>
> Der Amazon Inspector SBOM Generator generiert die `Package.resolved` Datei nicht. Wenn Ihr Projekt noch keine `Package.resolved` Datei enthält, müssen Sie diese vor dem Scannen generieren, indem Sie sie `swift package resolve` in Ihrem Projektverzeichnis ausführen.

Unterstützte Artefakte

- `Package.resolved`(Schemaversion 1)
- `Package.resolved`(Schemaversion 2)
- `Package.resolved`(Schemaversion 3)

Schlüssel-Features

- Extrahiert Abhängigkeitsnamen und gesperrte Versionen aus angehefteten Paketen
- Leitet den Paket-Namespace von der URL des Quell-Repositorys ab (zum Beispiel) `github.com/Alamofire`
- Überspringt nur Branch-Pins, für die es keine aufgelöste Version gibt
- Filtert lokale Quellcodeverwaltungs- und Dateisystemabhängigkeiten heraus
- Schließt `Package.resolved` Dateien innerhalb von `.build/` Verzeichnissen aus, um das Sammeln von Abhängigkeiten aus dem Swift Package Manager-Build-Cache zu verhindern

> **Note**
>
> Die `Package.resolved` Datei ist plattformunabhängig. Der Amazon Inspector SBOM Generator erkennt diese Datei rekursiv aufLinux, und. macOS Windows

Beispiel **`Package.resolved`** Datei (v2)

Das Folgende ist ein Beispiel für eine Version `Package.resolved` 2-Datei.

```
{
  "pins" : [
    {
      "identity" : "alamofire",
      "kind" : "remoteSourceControl",
      "location" : "https://github.com/Alamofire/Alamofire.git",
      "state" : {
        "revision" : "a1b2c3...",
        "version" : "5.8.1"
      }
    },
    {
      "identity" : "swift-argument-parser",
      "kind" : "remoteSourceControl",
      "location" : "https://github.com/apple/swift-argument-parser.git",
      "state" : {
        "revision" : "d4e5f6...",
        "version" : "1.3.0"
```

```
      }
    }
  ],
  "version" : 2
}
```

Beispiel **`Package.resolved`** Datei (v1)

Das Folgende ist ein Beispiel für eine Version `Package.resolved` 1-Datei.

```
{
  "object": {
    "pins": [
      {
        "package": "Alamofire",
        "repositoryURL": "https://github.com/Alamofire/Alamofire.git",
        "state": {
          "branch": null,
          "revision": "a1b2c3...",
          "version": "5.8.1"
        }
      }
    ]
  },
  "version": 1
}
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine Swift Package Manager-Abhängigkeit.

```
pkg:swift/github.com/Alamofire/Alamofire@5.8.1
```

> **Note**
>
> Jede dieser Dateien erzeugt eine Ausgabe, die eine Paket-URL enthält. Diese URL kann verwendet werden, um Informationen zu Softwarepaketen bei der Generierung einer

Softwareliste anzugeben. Sie kann in die ScanSbomAPI aufgenommen werden. Weitere Informationen finden Sie unter package-url auf der GitHub Website.

Nicht unterstützte Artefakte

In diesem Abschnitt werden nicht unterstützte Artefakte beschrieben.

Java

Der Amazon Inspector SBOM Generator Generator unterstützt nur die Erkennung von Sicherheitslücken für Abhängigkeiten, die aus dem Maven Mainstream-Repository stammen. Private oder benutzerdefinierte Maven Repositorys wie Red Hat Maven und werden nicht Jenkins unterstützt. Stellen Sie für eine genaue Erkennung von Sicherheitslücken sicher, dass Java Abhängigkeiten aus dem Maven Mainstream-Repository abgerufen werden. Abhängigkeiten von anderen Repositorys werden bei Schwachstellenscans nicht berücksichtigt.

JavaScript

Esbuild-Pakete

Bei esbuild minimierten Paketen unterstützt der Amazon Inspector SBOM Generator keinen Abhängigkeitsscan für Projekte mit. esbuild Quellzuordnungen, die von generiert wurdenesbuild, enthalten nicht genügend Metadaten (Namen und Versionen von Abhängigkeiten), die für eine genaue Generierung erforderlich sind. Sbomgen Um zuverlässige Ergebnisse zu erzielen, scannen Sie vor dem Bündelungsprozess die ursprünglichen Projektdateien`package-lock.json`, z. B. die Dateien `node_modules/directory` und.

package.json

Der Amazon Inspector SBOM Generator unterstützt das Scannen der package.json-Datei auf Stammebene nicht nach Abhängigkeitsinformationen. Diese Datei spezifiziert nur Paketnamen und Versionsbereiche, enthält jedoch keine vollständig aufgelösten Paketversionen. Verwenden Sie `package.json` oder andere Sperrdateien wie `yarn.lock` und, die aufgelöste Versionen enthaltenpnpm`.lock`, um genaue Scanergebnisse zu erzielen.

Punktnetz

Wenn Sie schwebende Versionen oder Versionsbereiche in verwendenPackageReference, wird es schwieriger, die genaue Paketversion zu ermitteln, die in einem Projekt verwendet wird, ohne

die Paketauflösung durchzuführen. Schwebende Versionen und Versionsbereiche ermöglichen es Entwicklern, anstelle einer festen Version eine Reihe akzeptabler Paketversionen anzugeben.

Gehen Sie zu Binärdateien

Der Amazon Inspector SBOM Generator scannt keine Go Binärdateien, die mit Build-Flags erstellt wurden, die so konfiguriert sind, dass sie die Build-ID ausschließen. Diese Build-Flags verhindern, dass Amazon Inspector SBOM Generator die Binärdatei genau ihrer ursprünglichen Quelle zuordnet. Unklare Go Binärdateien werden nicht unterstützt, da Paketinformationen nicht extrahiert werden können. Stellen Sie sicher, dass Go Binärdateien mit Standardeinstellungen, einschließlich der Build-ID, erstellt wurden, um genaue Abhängigkeiten zu überprüfen.

Rust-Binärdateien

Der Amazon Inspector SBOM Generator scannt nur Binärdateien, wenn die Rust Binärdateien mit der Cargo-Auditable Library erstellt wurden. RustBinärdateien, die diese Bibliothek nicht verwenden, verfügen nicht über die notwendigen Metadaten für eine genaue Extraktion von Abhängigkeiten. Der Amazon Inspector SBOM Generator extrahiert die kompilierte Rust Toolchainversion ab Version Rust 1.7.3, jedoch nur für Binärdateien in einer Umgebung. Linux Für umfassende Scans sollten Sie Binärdateien mithilfe von cargo-auditable erstellenRust. Linux

> **Note**
>
> Die Erkennung von Sicherheitslücken für die Rust Toolchain selbst wird nicht unterstützt, selbst wenn die Version der Toolchain extrahiert wurde.

Umfassende Ökosystemsammlung von Amazon Inspector SBOM Generator

Der Amazon Inspector SBOM Generator ist ein Tool zum Erstellen einer Software-Stückliste (SBOM) und zum Durchführen von Sicherheitslücken nach unterstützten Paketen aus Betriebssystemen und Programmiersprachen. Er unterstützt das Scannen verschiedener Ökosysteme außerhalb der Kernbetriebssysteme und gewährleistet so eine robuste und detaillierte Analyse der Infrastrukturkomponenten. Durch die Generierung einer SBOM können Sie die Zusammensetzung moderner Technologie-Stacks verstehen, Schwachstellen in Ökosystemkomponenten identifizieren und Einblicke in Software von Drittanbietern gewinnen.

Unterstützte Ökosysteme

Die Ökosystemsammlung erweitert die SBOM-Generierung über Pakete hinaus, die über Betriebssystem-Paketmanager installiert wurden. Dies erfolgt durch die Sammlung von Anwendungen, die mit alternativen Methoden, wie z. B. manueller Installation, bereitgestellt werden. Diese Anwendungssammler werden unter der `extra-ecosystems` Scanner-Gruppe ausgeführt, die standardmäßig für Container- und Localhost-Scans verwendet wird. Fügen Sie für andere Scantypen einen Scanner mit der `--additional-scanners` Option hinzu. Weitere Informationen finden Sie unter Verwenden des Amazon Inspector SBOM-Generators. Der Amazon Inspector SBOM Generator unterstützt das Scannen für die folgenden Ökosysteme:

Ökosysteme	Anwendungen
7-Zip	7-ZipArchiver (Version 21.07 und höher)
AI agent skills	Claude Cursor Fast-Agent OpenClaw Cross-client
AI IDE extensions	Continue Cody Cline
Amazon Q Developer	Amazon Q Developer CLI Amazon Q VS Code extension Amazon Q JetBrains plugin
Anthropic	Claude Code CLI Claude Desktop

Ökosysteme	Anwendungen
Anysphere	Cursor
Apache	Apache Cassandra Apache httpd Apache Struts Apache tomcat
Atlassian	Jira Core Confluence Jira Software Jira Service Management
Codeium	Windsurf
Conda	Miniconda-Umgebungen Anaconda-Umgebungen Miniforge-Umgebungen Mambaforge-Umgebungen
Curl	Curl Libcurl
Docker binaries	docker dockerd containerd runc
Elasticsearch	Elasticsearch

Ökosysteme	Anwendungen
Generic AI/ML Models	`.gguf` – GGUF `.safetensors` – Safetensors `.onnx` – ONNX `.pt` – PyTorch `.pth` – PyTorch `.h5` – Keras / HDF5
GitHub	GitHub Copilot CLI GitHub Copilot VS Code extension GitHub Copilot JetBrains plugin
Google	Chrome
HuggingFace	HuggingFace CLI Models Cache
Jan	Jan
Java	JDK JRE Amazon Corretto
Jenkins	Jenkins(Version 2.400.* und höher)
Kiro	Kiro CLI
LM Studio	LM Studio

Ökosysteme	Anwendungen
MariaDB und MySQL	MariaDB Server(10.6+, 11.x, 12.x) Oracle MySQL Server(8,0, 8,4, 9,4+) MySQL NDB Cluster
Metabase	Metabase
Microsoft applications	PowerShell NuGet CLI Visual Studio Code Microsoft Edge SharePoint Server Microsoft Defender Exchange Server Visual Studio .NET Core Runtime .NET Framework ASP.NET Core Runtime Microsoft Teams Outlook for Windows Microsoft Office Microsoft 365
Microsoft SQL Server	Microsoft SQL Server
MongoDB	MongoDB Server(7,0+, 8,0+)

Ökosysteme	Anwendungen
Mozilla	Firefox Firefox ESR Thunderbird
Nginx	Nginx
Node	Node
Node.JS	node
Nomic AI	GPT4All
Ollama	Ollama Ollama Model Collector
OpenSSH	OpenSSH(Versionen 9 und 10)
OpenSSL	OpenSSL
Oracle	Oracle Database Server
PHP	PHP(Version 8.1 und höher)
Redis	Redis(Version 7.2 und höher)
TabbyML	Tabby
WordPress	core plugin theme
Zed Industries	Zed

7-Zip Sammlung von Ökosystemen

Unterstützte Anwendungen

- 7 Zip-Archiver (Version 21.07 oder höher)

Schlüssel-Features

- Untersucht 7-Zip Binärdateien, um die eingebetteten Versionsinformationen zu extrahieren.

> **Note**
>
> Insbesondere sucht es nach dem Wert der Produktversion aus der Binärdatei.

Unterstützte Plattformen — Windows

- `C:/Program Files/7-Zip/7z.exe`
- `C:/Program Files/7-Zip/7za.exe`
- `C:/Program Files/7-Zip/7zz.exe`
- `C:/Program Files/7-Zip/7zr.exe`
- `C:/Program Files (x86)/7-Zip/7z.exe`
- `C:/Program Files (x86)/7-Zip/7za.exe`
- `C:/Program Files (x86)/7-Zip/7zz.exe`
- `C:/Program Files (x86)/7-Zip/7zr.exe`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für7-Zip.

```
pkg:generic/7zip/7zip@25.01
```

Fähigkeiten von KI-Agenten Sammlung von Ökosystemen

Unterstützte Anwendungen

- Claude
- OpenClaw
- Cursor
- Fast-Agent
- Cross-client

Schlüssel-Features

- Erkennt `SKILL.md` Dateien in Standard-Skill-Verzeichnissen für unterstützte Fähigkeiten von KI-Agenten.
- Unterstützt den mandantenübergreifenden Standardpfad (`~/.agents/skills/`) und agentenspezifische Pfade (z. B.). `~/.claude/skills/`
- Fähigkeiten sind nicht versioniert. Jeder Skill wird durch den SHA-256 Hash seiner `SKILL.md` Datei identifiziert.

Unterstützte Plattformen

Für Localhost-Scans scannt der Amazon Inspector SBOM Generator die folgenden Standardpfade. Bei Container-, Verzeichnis- und Archivscans werden `SKILL.md` Dateien überall im Dateisystem erkannt.

Linux

- `/home/<user>/.agents/skills/`
- `/home/<user>/<agent>/skills/`

macOS

- `/Users/<user>/.agents/skills/`
- `/Users/<user>/<agent>/skills/`

Windows

- `C:\Users\<user>\.agents\skills\`
- `C:\Users\<user>\<agent>\skills\`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für einen KI-Agenten-Skill.

```
# Claude skill
pkg:generic/agentskills/claude/web-design-guidelines

# Cross-client skill
pkg:generic/agentskills/cross-client/web-design-guidelines
```

AI-IDE-Erweiterungen Sammlung von Ökosystemen

Der `ai-ide-extensions` Collector ist ein einziger konsolidierter Scanner, der in Visual Studio Code JetBrains IDEs installierte Erweiterungen des KI-Codierungsassistenten erkennt. Entdeckte Erweiterungen werden als CycloneDX Komponenten mit dem `application` Typ und einem `generic` Paket-URL-Typ gemeldet.

Unterstützte Anwendungen

- Continue
- Cody(vonSourcegraph)
- Cline

Schlüssel-Features

- Erkennt die Erweiterung für beide unterstützten IDE-Familien. Bei Visual Studio Code Erweiterungen wird die Version aus der `package.json` Datei der Erweiterung gelesen. Bei JetBrains Plugins wird die Version aus der `plugin.xml` Datei gelesen.
- Jede unterstützte Erweiterung wird anhand ihrer IDE-Erweiterungs-ID (Visual Studio Code) oder ihres Plugin-Verzeichnisses (JetBrains) identifiziert:
  - Continue— Visual Studio Code Erweiterungs-ID`continue.continue-*`, JetBrains Plugin-Verzeichnis `continue`

- Cody— Visual Studio Code Erweiterungs-ID `sourcegraph.cody-ai-*`
- Cline— Visual Studio Code Erweiterungs-ID `saoudrizwan.claude-dev-*`

Unterstützte Plattformen

Cross-platform. Der Amazon Inspector SBOM Generator scannt überall dort, wo die IDE ihre Erweiterungen speichert.

Name des Scanners

`ai-ide-extensions`

Beispiel für eine PURL

Im Folgenden finden Sie Beispielpaket-URLs für die unterstützten AI-IDE-Erweiterungen.

```
# Continue
pkg:generic/continue-dev/continue@<version>

# Cody
pkg:generic/sourcegraph/cody@<version>

# Cline
pkg:generic/cline/cline@<version>
```

Amazon Q Developer Sammlung von Ökosystemen

Unterstützte Anwendungen

- Amazon Q Developer CLI
- Amazon Q VS Code extension
- Amazon Q JetBrains plugin

Schlüssel-Features

- Untersucht die q CLI-Binärdatei auf die eingebettete `AmazonQ-For-CLI` Identitätszeichenfolge und extrahiert die Version aus den Metadaten des AWS Tooling-Benutzeragenten.
- Analysiert bei IDE-Erweiterungen `package.json` (VS Code) und `META-INF/plugin.xml` (JetBrains), um die Version zu extrahieren.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux und macOS

- `~/.local/bin/q`
- `/usr/local/bin/q`
- `~/.vscode/extensions/amazonwebservices.amazon-q-vscode-*/`
- `<IDE>/plugins/amazon-q/META-INF/plugin.xml`

Windows

- `%APPDATA%\npm\q.cmd`
- `%USERPROFILE%\.vscode\extensions\amazonwebservices.amazon-q-vscode-*\`
- `%APPDATA%\JetBrains\<IDE>\plugins\amazon-q\`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürAmazon Q Developer.

```
Sample PURL: pkg:generic/amazon/amazon-q@1.19.7?distro=linux
```

Apache Sammlung von Ökosystemen

Dieser Abschnitt enthält Einzelheiten zu Apache CassandraApache httpd,Apache Struts, und Apache tomcat Anwendungen.

Apache Cassandra

Unterstützte Anwendungen

- Apache Cassandra

Schlüssel-Features

- Erkennt `apache-cassandra-<specific.version>.jar` Dateien und entpackt sie, um die Versionszeichenfolge aus dem `Implementation-Version` Eintrag in der Datei zu extrahieren. `META-INF/MANIFEST.MF`

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/share/cassandra/lib/`
- `/opt/cassandra/lib/`

macOS

- `/opt/homebrew/Cellar/cassandra/`
- `/usr/local/Cellar/cassandra/`

Windows

- `/Program Files/cassandra/lib/`
- `/Program Files/apache-cassandra/lib/`
- `/Program Files (x86)/cassandra/lib/`
- `/Program Files (x86)/apache-cassandra/lib/`

Beispiel **Apache-Cassandra-spezifische Version. < > jar/META- INF/MANIFEST.MF** file

Das Folgende ist ein Beispiel für Inhalte in einer `apache-cassandra-<specific.version>.jar/META-INF/MANIFEST.MF` Datei.

```
//truncated
```

```
Manifest-Version: 1.0
Implementation-Title: Cassandra
Implementation-Version: 4.1.3
Implementation-Vendor: Apache

//truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `Apache Cassandra` Anwendung.

```
Sample PURL: pkg:generic/apache/cassandra@4.1.3
```

Apache httpd

Unterstützte Anwendungen

- Apache httpd

> **Note**
>
> Die Bewertung der Sicherheitslücke gilt nur für Apache httpd Version 2.0 und höher.

Schlüssel-Features

- Analysiert die `/include/ap_release.h` Datei, um Installationsmakros zu extrahieren, die Hauptkennungen, Nebenkennungen und Patchkennungen enthalten.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Unix

- `/usr/local/apache2/include/`

Windows

- `/Apache24/include/`
- `/Program Files/Apache24/include/`
- `/Program Files (x86)/Apache24/include/`

Beispiel **`ap_release.h`** file

Das Folgende ist ein Beispiel für den Inhalt einer Datei. `ap_release.h`

```
//truncated

#define AP_SERVER_BASEVENDOR "Apache Software Foundation"
#define AP_SERVER_BASEPROJECT "Apache HTTP Server"
#define AP_SERVER_BASEPRODUCT "Apache"

#define AP_SERVER_MAJORVERSION_NUMBER 2
#define AP_SERVER_MINORVERSION_NUMBER 4
#define AP_SERVER_PATCHLEVEL_NUMBER   1
#define AP_SERVER_DEVBUILD_BOOLEAN    0

//truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `Apache httpd` Anwendung.

```
Sample PURL: pkg:generic/apache/httpd@2.4.1
```

Apache Struts

Unterstützte Anwendungen

- Apache Struts

Schlüssel-Features

- Erkennt `struts2-core-<specific.version>.jar` Dateien (die normalerweise im `WEB-INF/lib/` Verzeichnis einer Webanwendung auf einem Java Anwendungsserver bereitgestellt werden) und entpackt sie, um die Versionszeichenfolge aus dem `Implementation-Version` Eintrag in der Datei zu extrahieren. `META-INF/MANIFEST.MF`

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gängigen Java Anwendungsserverpfaden:

Linux

- `/opt/tomcat/`
- `/usr/share/tomcat/`
- `/var/lib/tomcat/`
- `/usr/local/tomcat/`
- `/opt/wildfly/`
- `/opt/jboss/`
- `/opt/jetty/`

Windows

- `/Program Files/Apache Software Foundation/`
- `/Program Files (x86)/Apache Software Foundation/`

Beispiel **`struts2-core – spezifische.version. < > jar/META- INF/MANIFEST.MF`** file

Das Folgende ist ein Beispiel für Inhalte in einer `struts2-core-<specific.version>.jar/META-INF/MANIFEST.MF` Datei.

```
//truncated

Manifest-Version: 1.0
```

```
Implementation-Title: Struts 2 Core
Implementation-Version: 6.4.0
Implementation-Vendor: Apache Software Foundation

//truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `Apache Struts` Anwendung.

```
Sample PURL: pkg:generic/apache/struts@6.4.0
```

Apache Tomcat

Unterstützte Anwendungen

- Apache tomcat

> **Note**
> Die Bewertung der Sicherheitslücke gilt nur für Apache tomcat Version 9.0 und höher.

Schlüssel-Features

- Entpackt die `catalina.jar` Datei, um die Installationsmakros in der `META-INF/MANIFEST.MF` Datei zu extrahieren, die die Versionszeichenfolge enthält.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `/opt/tomcat/lib/`

- `/usr/share/tomcat/lib`
- `/var/lib/tomcat/lib/`

macOS

- `/Library/Tomcat/lib/`
- `/usr/local/tomcat/lib`

Windows

- `/Program Files/Apache Software Foundation`
- `/Program Files (x86)/Apache Software Foundation/`

Beispiel **`Catalina. jar/META- INF/MANIFEST.MF`** file

Das Folgende ist ein Beispiel für Inhalte in einer `catalina.jar/META-INF/MANIFEST.MF` Datei.

```
//truncated

Implementation-Title: Apache Tomcat
Implementation-Vendor: Apache Software Foundation
Implementation-Version: 10.1.31

//truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `Apache tomcat` Anwendung.

```
Sample PURL: pkg:generic/apache/tomcat@10.1.31
```

Atlassian Sammlung von Ökosystemen

Dieser Abschnitt enthält Einzelheiten zu Atlassian Serverprodukten und -anwendungen.

Atlassian Serverprodukte

Unterstützte Anwendungen

- Jira Core
- Confluence

Schlüssel-Features

- Jira Core— Analysiert die Maven-POM-Eigenschaften von, um Versionsinformationen `atlassian-jira-webapp` zu extrahieren.
- Confluence— Analysiert die Maven-POM-Eigenschaften von, um Versionsinformationen `confluence-webapp` zu extrahieren.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht nach Installationen in gängigen Installationspfaden:

Linux

- `/opt/atlassian/jira/atlassian-jira/META-INF/maven/com.atlassian.jira/atlassian-jira-webapp/pom.properties`
- `/opt/atlassian/confluence/confluence/META-INF/maven/com.atlassian.confluence/confluence-webapp/pom.properties`

Beispiel für eine PURL

Im Folgenden finden Sie Beispielpaket-URLs für Atlassian Serverprodukte.

```
// Jira Core
pkg:generic/atlassian/jira-core@10.0.1?distro=linux

// Confluence
pkg:generic/atlassian/confluence@9.2.7?distro=linux
```

Atlassian-Anwendungen

Unterstützte Anwendungen

- Jira Software
- Jira Service Management

Schlüssel-Features

- Jira Software— Erkennt die Version über `jira-software-application` JAR und extrahiert die Version aus den POM-Eigenschaften von Maven.
- Jira Service Management— Erkennt über `jira-servicedesk-application` JAR und extrahiert die Version aus den Maven-POM-Eigenschaften.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht nach Installationen in gängigen Installationspfaden:

Linux

- `/opt/atlassian/jira/atlassian-jira/WEB-INF/application-installation/jira-software-application/jira-software-application-*.jar`
- `/opt/atlassian/jira/atlassian-jira/WEB-INF/application-installation/jira-servicedesk-application/jira-servicedesk-application-*.jar`

Beispiel für eine PURL

Im Folgenden finden Sie Beispielpaket-URLs für Atlassian Anwendungen.

```
// Jira Software
pkg:generic/atlassian/jira-software@10.3.9?distro=linux

// Jira Service Management
pkg:generic/atlassian/jira-service-management@10.3.9?distro=linux
```

Claude Code Sammlung von Ökosystemen

Unterstützte Anwendungen

- Claude Code CLI(Anthropisch)

Schlüssel-Features

- Claude Codewird als npm-Paket () `@anthropic-ai/claude-code` verteilt. Der Scanner erkennt die `claude` CLI-Binärdatei und löst sie `package.json` relativ dazu auf, wobei das `<prefix>/bin/<binary>` Standard-NPM-Layout verwendet wird. `<prefix>/lib/node_modules/@anthropic-ai/claude-code/package.json`
- Die Version wird aus dem Feld von gelesen. `Version: package.json`

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux und macOS

- `/usr/bin/claude`
- `/usr/local/bin/claude`
- `~/.local/bin/claude`
- `~/.npm-global/bin/claude`

Windows

- `%APPDATA%\npm\claude.cmd`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürClaude Code.

```
Sample PURL: pkg:generic/anthropic/claude-code@1.0.5?distro=linux
```

Conda Sammlung von Ökosystemen

Unterstützte Anwendungen

- Miniconda-Umgebungen
- Anaconda-Umgebungen
- Miniforge-Umgebungen
- Mambaforge-Umgebungen

Schlüssel-Features

- Analysiert `conda-meta/*.json` Paketdatensätze, die von conda und mamba in jeder Conda-Umgebung geschrieben wurden.
- Extrahiert den Paketnamen, die Version, die Build-Zeichenfolge, das Unterverzeichnis (Zielplattform) und den Archivtyp (`condaodertar.bz2`) gemäß CEP 26.
- Generiert conda Paket-URLs.
- Filtert macOS Resource-Fork-Dateien (`._*`) heraus und überspringt Datensätze mit einem leeren Namen oder einer leeren Version.

Unterstützte Plattformen

Bei Localhost-Scans scannt der Amazon Inspector SBOM Generator Conda-Meta-Verzeichnisse in den folgenden Installer-Standardpfaden und benutzerspezifischen Pfaden. Bei Container-, Verzeichnis- und Archivscans erkennt der Amazon Inspector SBOM Generator `conda-meta/*.json` Datensätze an beliebiger Stelle in der Baumstruktur.

Linux

- `/opt/conda`
- `/opt/miniconda3`, `/opt/anaconda3`, `/opt/miniforge3`
- `/root/miniconda3`, `/root/anaconda3`, `/root/miniforge3`
- `/home/<user>/miniconda3`, `/home/<user>/anaconda3`, `/home/<user>/miniforge3`
- `~/.conda/envs/<env>`

macOS

- `/opt/miniconda3`, `/opt/anaconda3`
- `/opt/homebrew/anaconda3`, `/usr/local/anaconda3`
- `/opt/homebrew/Caskroom/miniconda/base`, `/opt/homebrew/Caskroom/miniforge/base`
- `/Users/<user>/miniconda3`, `/Users/<user>/anaconda3`, `/Users/<user>/miniforge3`
- `~/.conda/envs/<env>`

Windows

- `C:\ProgramData\miniconda3`, `C:\ProgramData\anaconda3`, `C:\ProgramData\miniforge3`
- `C:\Users\<user>\miniconda3`, `C:\Users\<user>\anaconda3`, `C:\Users\<user>\miniforge3`
- `C:\Users\<user>\.conda\envs\<env>`

Beispiel **`Conda-Meta`** Pfad aufzeichnen

Im Folgenden finden Sie einen Beispielpfad für einen conda Paketdatensatz.

```
/opt/miniforge3/conda-meta/numpy-1.24.0-py311h64a7726_0.json
```

Beispiel PURL

Im Folgenden finden Sie Beispielpaket-URLs für ein plattformspezifisches conda Paket und ein Paket. `noarch` conda

```
pkg:conda/numpy@1.24.0?build=py311h64a7726_0&subdir=linux-64&type=conda
pkg:conda/flask@3.0.0?build=pyhd8ed1ab_0&subdir=noarch&type=conda
```

Curl Sammlung von Ökosystemen

Dieser Abschnitt enthält Einzelheiten zu Curl und Libcurl Anwendungen.

Curl

Unterstützte Anwendungen

- Curl

Unterstützte Plattformen

- Unix – Linux und macOS
  - /usr/local/bin/curl

Hauptmerkmale — Curl

- Untersucht curl Binärdateien, um die eingebetteten Versionsinformationen zu extrahieren.

> **Note**
>
> Insbesondere sucht es nach Versionszeichenfolgen im `.rodata` Abschnitt mit ausführbaren Binärdateien (für ELF-Binärdateien unter Linux), im `.rdata` Abschnitt (für PE-Binärdateien unter Windows) oder im Abschnitt __cstring (für MachO-Binärdateien unter macOS).

Curl-Versionszeichenfolge

Das Folgende ist ein Beispiel für eine Versionszeichenfolge, die in eine Curl Binärdatei eingebettet ist:

```
curl/8.14.1
```

Version `8.14.1` wird aus der Zeichenfolge extrahiert, um die `Curl` Version zu identifizieren.

Beispiel PURL (Curl)

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `Curl` Versionsdatei.

```
Sample PURL: pkg:generic/curl/curl@8.14.1
```

Libcurl

Unterstützte Anwendungen

- Libcurl

Unterstützte Plattformen

- Unix – Linux und macOS
  - //usr/local/curlver.h bin/curl

Hauptmerkmale — Libcurl

- Überprüftcurlver.h, ob eingebettete Versionsinformationen für extrahiert werden sollen. Libcurl

> **Note**
>
> Insbesondere extrahiert es die Version aus den definierten `LIBCURL_VERSION_PATCH` Variablen `LIBCURL_VERSION_MAJOR``LIBCURL_VERSION_MINOR`, und.

Libcurl-Versionszeichenfolge

Das Folgende ist ein Beispiel für die Versionsvariablen in einer `curlver.h` Datei:

```
#define LIBCURL_VERSION_MAJOR 8
    #define LIBCURL_VERSION_MINOR 14
    #define LIBCURL_VERSION_PATCH 1
```

`8.14.1`Die Version wird aus diesen Zeilen extrahiert, um die `Libcurl` Version zu identifizieren.

Beispiel PURL (Libcurl)

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `Libcurl` Versionsdatei.

```
Sample PURL: pkg:generic/curl/libcurl@8.14.1
```

Cursor Sammlung von Ökosystemen

Unterstützte Anwendungen

- Cursor(vonAnysphere)

Schlüssel-Features

- Untersucht Installationen vonCursor, einem Electron auf KI basierenden Code-Editor vonAnysphere. Extrahiert die Version aus dem `resources/app/package.json` Electron-Manifest der Anwendung.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/share/cursor/`
- `/usr/bin/cursor`
- `~/.local/bin/cursor`

Windows

- `%LOCALAPPDATA%\Programs\cursor\`
- `<SystemDrive>\Program Files\cursor\`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürCursor.

```
Sample PURL: pkg:generic/cursor/cursor@0.42.3?distro=linux
```

Docker-Binärdateien Sammlung von Ökosystemen

Unterstützte Anwendungen

- docker (docker version 20.10.18+)
- dockerd (docker version 20.10.18+)
- containerd (docker version 20.10.18+)
- runc (docker version 29.0.0+)

> **Note**
>
> Die Bewertung der Sicherheitslücke gilt nur für die Docker Versionen 25.0.0+. Frühere Versionen von Docker Binärdateien können gesammelt werden, sie werden jedoch nicht mehr von unterstützt. Docker

Schlüssel-Features

- Untersucht Docker statische Binärdateien, um Versionsinformationen zu extrahieren.

Unterstützte Plattformen

Der SBOM-Generator sucht in den Installationspfaden, die in der Dokumentation zur statischen Binärinstallation von Docker empfohlen werden, nach Binärdateien:

Linux

- `/usr/bin/`

macOS

- `/usr/local/bin/`

Windows

- `C:/Program Files/Docker/`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürDocker.

```
pkg:generic/docker/docker@29.4.0
pkg:generic/docker/dockerd@29.4.0
pkg:generic/docker/containerd@2.2.2
pkg:generic/docker/runc@1.3.5
```

Elasticsearch Sammlung von Ökosystemen

Unterstützte Anwendungen

- Elasticsearch

> **Note**
>
> Die Bewertung der Sicherheitslücke bezieht sich nur auf Elasticsearch Version 7.17.0.

Schlüssel-Features

- Version— Entpackt die `elasticsearch-<specific.version>.jar` Datei, um Installationsmakros in `META-INF/MANIFEST.MF` Dateien zu extrahieren, die die Versionszeichenfolge enthalten. Elasticsearch

Unterstützte Plattformen

- Linux—`/etc/elasticsearch/lib`, und `/opt/elasticsearch/lib/` `/usr/share/elasticsearch/lib/`
- macOS – `/usr/local/var/lib/elasticsearch/lib/`
- Windows—`/elasticsearch/`,`/Program Files (x86)/Elastic/elasticsearch/lib/`, und `/Program Files/Elastic/elasticsearch/lib/`

Beispiel **`<Elasticsearch-spezifische > Version. jar/META- INF/MANIFEST.MF`** file

Das Folgende ist ein Beispiel für eine `elasticsearch-<specific.version>.jar/META-INF/MANIFEST.MF` Datei.

```
//truncated

Manifest-Version: 1.0
Module-Origin: git@github.com:elastic/elasticsearch.git
X-Compile-Elasticsearch-Version: 8.19.0-SNAPSHOT
X-Compile-Lucene-Version: 9.12.1
X-Compile-Elasticsearch-Snapshot: true

//truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `elasticsearch-<specific.version>.jar/META-INF/MANIFEST.MF` Datei.

```
pkg:generic/elastic/elasticsearch@8.19.0-SNAPSHOT
```

Generische AI/ML Modelle Sammlung von Ökosystemen

Unterstützte Anwendungen

- Lokal gespeicherte AI/ML Modelldateien in gängigen Frameworks und Tools

Schlüssel-Features

- Erkennt AI/ML Modelldateien anhand der Dateierweiterung: `.gguf.safetensors`,`.onnx`,`.pt`,`.pth`, und `.h5`.
- Scannt gängige Modellverzeichnisse, die von gängigen AI/ML Frameworks und lokalen Inferenztools verwendet werden.
- Erkannte Dateien werden als Komponenten mit dem `machine-learning-model` Komponententyp generiert.

> **Note**
>
> Für Komponenten, die von diesem Kollektor erkannt wurden, wird keine PURL generiert.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Modelldateien in gemeinsamen Pfaden:

Linux und macOS

- `~/.keras/models`
- `~/.cache/lm-studio/models`
- `~/.local/share/nomic.ai/GPT4All`
- `~/.jan/models`
- `~/.cache/llama.cpp`
- `~/.tabby/models`
- `~/.local/share/localai/models`
- `~/text-generation-webui/models`
- `~/ComfyUI/models`
- `~/stable-diffusion-webui/models`

GitHub Copilot Sammlung von Ökosystemen

Unterstützte Anwendungen

- GitHub Copilot CLI
- GitHub Copilot VS Code extension
- GitHub Copilot JetBrains plugin

Schlüssel-Features

- Erkennt Copilot drei Installationsoberflächen: npm-distributed CLI (löst `package.json` aus dem Binärpfad über das npm-Layout auf), VS Code-Erweiterung (analysiert `package.json` in `github.copilot-<version>` Verzeichnissen; schließt die separate `github.copilot-chat-*` Erweiterung aus) und JetBrains Plugin (analysiert das Element). `META-INF/plugin.xml` `<version>`

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux und macOS

- `/usr/bin/github-copilot-cli`
- `/usr/local/bin/github-copilot-cli`
- `~/.local/bin/github-copilot-cli`
- `~/.vscode/extensions/github.copilot-*/`
- `<IDE>/plugins/github-copilot/META-INF/plugin.xml`

Windows

- `%APPDATA%\npm\github-copilot-cli.cmd`
- `%USERPROFILE%\.vscode\extensions\github.copilot-*\`
- `%APPDATA%\JetBrains\<IDE>\plugins\github-copilot\`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürGitHub Copilot.

```
Sample PURL: pkg:generic/github/copilot@0.1.36?distro=linux
```

Google Sammlung von Ökosystemen

Unterstützte Anwendungen

- Google Chrome
- Puppeteer(unterstützt die Puppeteer-Bibliothek; Puppeteer-Core ist nicht enthalten)

> **Note**
>
> Puppeteerunterstützt die Puppenspieler-Bibliothek. PuppeteerKern ist nicht enthalten.

Unterstützte Artefakte

Amazon Inspector sammelt Google Chrome Informationen aus den folgenden Quellen:

- Die `chrome/VERSION` Datei (Build-Quelle)
- Die `chrome.exe` Datei (Windows ChromeInstallation)
- Die `puppeteer` Datei (Installation)

Für jedes der unterstützten Artefakte Sbomgen analysiert und sammelt er entweder die chrome Datei oder die puppeteer Datei. Bei puppeteer Installationen wird die entsprechende Chromium Version auf der Grundlage der puppeteer Version gesammelt. Weitere Informationen finden Sie auf der Puppeteer-Website unter Unterstützte Browser.

Wenn eine der folgenden Umgebungsvariablen auf gesetzt ist`true`, wird die Auswertung übersprungen und der `skip_chromium_download=true` Qualifier wird der Paket-URL hinzugefügt. Puppeteer

- `PUPPETEER_SKIP_CHROMIUM_DOWNLOAD`
- `PUPPETEER_CHROME_SKIP_DOWNLOAD`
- `PUPPETEER_SKIP_DOWNLOAD`

Beispiel **`chrome/VERSION`** Versionsdatei

Das Folgende ist ein Beispiel für die `chrome/VERSION` Versionsdatei.

```
MAJOR=130
MINOR=0
BUILD=6723
PATCH=58
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `chrome/VERSION` Versionsdatei.

```
Sample PURL: pkg:generic/google/chrome@131.0.6778.87
```

Beispiel **`Puppenspieler`** Versionsdatei

Das Folgende ist ein Beispiel für die `puppeteer` Versionsdatei.

```
{
"name": "puppeteer",
"version": "23.9.0",
"description": "A high-level API to control headless Chrome over the DevTools
 Protocol",
"keywords": [
  "puppeteer",
  "chrome",
  "headless",
  "automation"
]
}
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine `puppeteer` Versionsdatei.

```
Sample PURL: pkg:generic/google/puppeteer@23.9.0
```

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL mit Skip-Qualifier für eine `puppeteer` Versionsdatei.

```
pkg:generic/google/puppeteer@22.15.0?distro=linux&skip_chromium_download=true
```

Homebrew Sammlung von Ökosystemen

Unterstützte Anwendungen

- HomebrewPaketmanager

Schlüssel-Features

- Erkennt Homebrew -verwaltete Pakete, indem es alle Formeln `INSTALL_RECEIPT.json` im Verzeichnis liest Homebrew `Cellar`
- Scannt die `Cellar` Standardspeicherorte (`/usr/local/Cellar/opt/homebrew/Cellar`, und`/home/linuxbrew/.linuxbrew/Cellar`) sowie den in der `HOMEBREW_CELLAR` Umgebungsvariablen angegebenen Speicherort
- Generiert Homebrew Paket-URLs mit dem `brew` Typ

Unterstützte Plattformen

- macOS
- Linux

Beispielpfad

Im Folgenden finden Sie ein Beispiel für einen Empfangspfad für die Homebrew Paketinstallation.

```
/opt/homebrew/Cellar/<name>/<version>/INSTALL_RECEIPT.json
```

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein Homebrew -verwaltetes Paket.

```
pkg:brew/<name>@<version>
```

HuggingFace Sammlung von Ökosystemen

Unterstützte Anwendungen

- HuggingFace`hf`CLI

Schlüssel-Features

- Extrahiert lokal zwischengespeicherte AI/ML Modelle, die installiert wurden von HuggingFace
- Generiert HuggingFace Paket-URLs
- Modelle, die mit heruntergeladen wurden, `hf download --local-dir` werden derzeit nicht unterstützt

Beispielpfad

Das Folgende ist ein Beispiel für einen zwischengespeicherten HuggingFace Modellpfad.

```
/home/ec2-user/.cache/huggingface/hub/models--MiniMaxAI--MiniMax-M2.5/snapshots/<hash>
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein HuggingFace Modell. Der Komponententyp ist`machine-learning-model`.

```
pkg:huggingface/MiniMaxAI/MiniMax-M2.5@<hash>
```

Java Sammlung von Ökosystemen

Unterstützte Anwendungen

- Oracle JDK
- Oracle JRE
- Amazon Corretto

Schlüssel-Features

- Extrahiert die Zeichenfolge der Java Installation.
- Identifiziert den Verzeichnispfad, der die Java Laufzeit enthält.
- Identifiziert den Anbieter als Oracle JDKOracle JRE, undAmazon Corretto.

Der Amazon Inspector SBOM Generator sucht auf den folgenden Installationspfaden und Plattformen nach Java Installationen:

- macOS: `/Library/Java/JavaVirtualMachines`
- Linux 32-bit: `/usr/lib/jvm`
- Linux 64-bit: `/usr/lib64/jvm`
- Linux (generic): `/usr/java and /opt/java`

Beispiel Java Versionsinformationen

Das Folgende ist ein Beispiel für eine Oracle Java Veröffentlichung.

```
// Amazon Corretto
IMPLEMENTOR="Amazon.com Inc."
IMPLEMENTOR_VERSION="Corretto-17.0.11.9.1"
JAVA_RUNTIME_VERSION="17.0.11+9-LTS"
JAVA_VERSION="17.0.11"
```

```
JAVA_VERSION_DATE="2024-04-16"
LIBC="default"
MODULES="java.base java.compiler java.datatransfer java.xml java.prefs java.desktop
 java.instrument java.logging java.management java.security.sasl java.naming
 java.rmi java.management.rmi java.net.http java.scripting java.security.jgss
 java.transaction.xa java.sql java.sql.rowset java.xml.crypto java.se java.smartcardio
 jdk.accessibility jdk.internal.jvmstat jdk.attach jdk.charsets jdk.compiler
 jdk.crypto.ec jdk.crypto.cryptoki jdk.dynalink jdk.internal.ed jdk.editpad
 jdk.hotspot.agent jdk.httpserver jdk.incubator.foreign jdk.incubator.vector
 jdk.internal.le jdk.internal.opt jdk.internal.vm.ci jdk.internal.vm.compiler
 jdk.internal.vm.compiler.management jdk.jartool jdk.javadoc jdk.jcmd jdk.management
 jdk.management.agent jdk.jconsole jdk.jdeps jdk.jdwp.agent jdk.jdi jdk.jfr jdk.jlink
 jdk.jpackage jdk.jshell jdk.jsobject jdk.jstatd jdk.localedata jdk.management.jfr
 jdk.naming.dns jdk.naming.rmi jdk.net jdk.nio.mapmode jdk.random jdk.sctp
 jdk.security.auth jdk.security.jgss jdk.unsupported jdk.unsupported.desktop
 jdk.xml.dom jdk.zipfs"
OS_ARCH="x86_64"
OS_NAME="Darwin"
SOURCE=".:git:7917f11551e8+"

// JDK
IMPLEMENTOR="Oracle Corporation"
JAVA_VERSION="19"
JAVA_VERSION_DATE="2022-09-20"
LIBC="default"
MODULES="java.base java.compiler java.datatransfer java.xml java.prefs java.desktop
 java.instrument java.logging java.management java.security.sasl java.naming
 java.rmi java.management.rmi java.net.http java.scripting java.security.jgss
 java.transaction.xa java.sql java.sql.rowset java.xml.crypto java.se java.smartcardio
 jdk.accessibility jdk.internal.jvmstat jdk.attach jdk.charsets jdk.zipfs jdk.compiler
 jdk.crypto.ec jdk.crypto.cryptoki jdk.dynalink jdk.internal.ed jdk.editpad
 jdk.hotspot.agent jdk.httpserver jdk.incubator.concurrent jdk.incubator.vector
 jdk.internal.le jdk.internal.opt jdk.internal.vm.ci jdk.internal.vm.compiler
 jdk.internal.vm.compiler.management jdk.jartool jdk.javadoc jdk.jcmd jdk.management
 jdk.management.agent jdk.jconsole jdk.jdeps jdk.jdwp.agent jdk.jdi jdk.jfr jdk.jlink
 jdk.jpackage jdk.jshell jdk.jsobject jdk.jstatd jdk.localedata jdk.management.jfr
 jdk.naming.dns jdk.naming.rmi jdk.net jdk.nio.mapmode jdk.random jdk.sctp
 jdk.security.auth jdk.security.jgss jdk.unsupported jdk.unsupported.desktop
 jdk.xml.dom"
OS_ARCH="x86_64"
OS_NAME="Darwin"
SOURCE=".:git:53b4a11304b0 open:git:967a28c3d85f"
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine Oracle Java Version.

```
Sample PURL:
# Amazon Corretto
pkg:generic/amazon/amazon-corretto@21.0.3
# Oracle JDK
pkg:generic/oracle/jdk@11.0.16
# Oracle JRE
pkg:generic/oracle/jre@20
```

Jenkins Sammlung von Ökosystemen

Unterstützte Anwendungen

- Jenkins-Core

> **Note**
>
> Die Bewertung der Sicherheitslücke bezieht sich auf Jenkins Version 2.400.* und höher.

Schlüssel-Features

- Extrahiert Versionsinformationen aus der `jenkins.war` Datei, indem die `META-INF/MANIFEST.M` Datei gelesen wird, die die Versionszeichenfolge enthält. Jenkins

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Jenkins-Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/share/jenkins/jenkins.war`
- usr/sharejava/jenkins//.war

macOS

- `/opt/homebrew/opt/jenkins-lts/libexec/jenkins.war`

Windows

- `/Program Files/Jenkins/Jenkins.war`
- `/Program Files (x86)/Jenkins/Jenkins.war`

Beispieldateien

Im Folgenden finden Sie Beispiele für `jenkins.war/META-INF/MANIFEST.MF` Dateien für verschiedene Versionen.

```
Manifest-Version: 1.0
Created-By: Maven WAR Plugin 3.4.0
Build-Jdk-Spec: 21
Implementation-Title: Jenkins war
Main-Class: executable.Main
Implementation-Version: 2.516.2
Jenkins-Version: 2.516.2
```

```
Manifest-Version: 1.0
Jenkins-Version: 2.414.1
Implementation-Title: Jenkins
Implementation-Version: 2.414.1
Built-By: kohsuke
Created-By: Apache Maven 3.8.6
```

Beispiel-PURLs

Im Folgenden finden Sie Paket-URLs für Version 2.516.2 der Jenkins LTS-Version und Version 2.414 der Automation-Server-Version. Jenkins

```
LTS: pkg:generic/jenkins/jenkins-core-lts@2.516.2.1
Regular: pkg:generic/jenkins/jenkins-core@2.414
```

Kiro CLI Sammlung von Ökosystemen

Unterstützte Anwendungen

- Kiro CLI(Amazon AI-Codierungsassistent)

Schlüssel-Features

- Untersucht die `kiro-cli` Binärdatei (oder`kiro-cli.appimage`), um eingebettete Versionsinformationen zu extrahieren.
- Sucht nach Versionszeichenfolgen im `.rodata` Abschnitt der ausführbaren Binärdatei (für ELF-Binärdateien aktiviertLinux) oder `__cstring` Abschnitt (für Mach-O Binärdateien aktiviertmacOS). Die Version wird aus dem Zeichenkettenmuster des AWS Tooling-Benutzeragenten extrahiert. `Version<X.Y.Z>/`

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `~/.local/bin/kiro-cli`
- `/usr/bin/kiro-cli`
- `~/Applications/kiro-cli.appimage`

macOS

- `/Applications/Kiro CLI.app/Contents/MacOS/kiro-cli`
- `~/.local/bin/kiro-cli`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürKiro CLI.

```
Sample PURL: pkg:generic/amazon/kiro@1.29.5?distro=linux
```

MariaDB und MySQL Sammlung von Ökosystemen

MariaDB

Unterstützte Anwendungen

- MariaDB Server(10.6+, 11.x, 12.x)

Schlüssel-Features

- Extrahiert Versionsinformationen aus Datenbankserver-Binärdateien und Header-Dateien mithilfe datenbankspezifischer Muster.
- Identifiziert den Verzeichnispfad, der die Datenbankserver-Installation enthält.
- Unterscheidet mithilfe der datengesteuerten Dateityperkennung automatisch zwischen MySQL Installationen MariaDB und Installationen.

Der SBOM-Generator sucht plattformübergreifend in gemeinsamen MariaDB Installationspfaden nach der Installation:

Linux

- `/usr/bin/mariadbd`
- `/usr/sbin/mariadbd`
- `/usr/local/bin/mariadbd`

macOS

- `/opt/homebrew/Cellar/mariadb/include/mysql/mariadb_version.h (MariaDB)`

Windows

- `C:/Program Files (x86)/MariaDB/include/mysql/mariadb_version.h (MariaDB)`
- `C:/Program Files/MariaDB/include/mysql/mariadb_version.h (MariaDB)`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für einen MariaDB Server.

```
# MariaDB Server

pkg:generic/mysql/mariadb-server@10.11.8
```

MySQL Sammlung von Ökosystemen

Unterstützte Anwendungen

- Oracle MySQL Server(8.0, 8.4, 9.4+)
- MySQL NDB Cluster

Schlüssel-Features

- Extrahiert Versionsinformationen aus Datenbankserver-Binärdateien und Header-Dateien mithilfe datenbankspezifischer Muster.
- Identifiziert den Verzeichnispfad, der die Datenbankserver-Installation enthält.
- Unterscheidet mithilfe der datengesteuerten Dateityperkennung automatisch zwischen MariaDB Installationen MySQL und Installationen.
- Identifiziert MySQL NDB Cluster Builds des Servers und meldet sie wie die `mysql-cluster` NDB Cluster Version.

Der SBOM-Generator sucht plattformübergreifend in gängigen MySQL Installationspfaden nach der Installation:

Linux

- `/usr/local/bin/mysqld`
- `/usr/bin/mysqld`
- `/usr/sbin/mysqld`

macOS

- `/usr/local/mysql/include/mysql_version.h (MySQL)`

Windows

- `C:/Program Files/MySQL/MySQL Server/include/mysql_version.h (MySQL)`
- `C:/Program Files (x86)/MySQL/MySQL Server/include/mysql_version.h (MySQL)`
- `C:/mysql-cluster-*-winx64/bin/mysqld.exe (MySQL NDB Cluster)`

Beispiel für eine PURL

Im Folgenden finden Sie Beispielpaket-URLs für einen MySQL Server und einen MySQL NDB Cluster Server.

```
# Oracle MySQL Server

pkg:generic/mysql/mysql-server@8.0.43

# MySQL NDB Cluster

pkg:generic/mysql/mysql-cluster@7.6.25
```

Microsoft-Anwendungen Sammlung von Ökosystemen

Die folgenden Microsoft-Anwendungen werden vom Amazon Inspector SBOM-Generator inventarisiert. Aufgrund von Einschränkungen in der Microsoft CVRF-API werden Erkennungen in der InspectorScan API nur für Versionen dieser Anwendungen unterstützt, die 2021 (oder später) veröffentlicht wurden. Die Ergebnisse werden Microsoft-KBs oder CVEs (sofern zutreffend) zugeordnet.

Unterstützte Microsoft-Anwendungen (2021+)

- PowerShell
- NuGet CLI
- Visual Studio Code
- Microsoft Edge
- SharePoint Server
- Microsoft Defender
- Exchange Server
- Visual Studio

- .NET Core Runtime
- .NET Framework
- ASP.NET Core Runtime
- Microsoft Teams
- Outlook for Windows
- Microsoft Office
- Microsoft 365

Schlüssel-Features

- PowerShell— Untersucht die `pwsh.exe` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- NuGet CLI— Untersucht die `nuget.exe` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- Visual Studio Code— Untersucht die `Code.exe` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- Microsoft Edge— Untersucht die `msedge.exe` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- SharePoint Server— Untersucht die `Microsoft.SharePoint.dll` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- Microsoft Defender— Untersucht die `MsMpEng.exe` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- Exchange Server— Untersucht die `Exsetup.exe` Datei, um die eingebetteten Versionsinformationen zu extrahieren.
- Visual Studio— Analysiert die `state.json` Datei, um die Versionszeichenfolge aus dem `catalogInfo.productDisplayVersion` Feld abzurufen.
- .NET Core Runtime— Sucht `Microsoft.NETCore.App.deps.json` in Installationspfaden nach Dateien und extrahiert die Versionszeichenfolge aus dem folgenden Dateipfadmuster.

  ```
  Microsoft.NETCore.App/<VERSION>/Microsoft.NETCore.App.deps.json
  ```

- .NET Framework— Analysiert die Windows-Registrierung und liest Dateimetadaten, um installierte.NET Framework-Versionen zu erkennen. Der Scanner überprüft den folgenden Registrierungsschlüssel und -wert sowie die folgenden Dateien.

- Registry Key(<VERSION_SUB_KEY>steht für die Version von.NET Framework, z. B. v2.0.50727v3.5, oderv4\Full)

```
HKLM\SOFTWARE\Microsoft\NET Framework Setup\NDP\<VERSION_SUB_KEY>
```

```
HKLM\SOFTWARE\Wow6432Node\Microsoft\NET Framework Setup\NDP\<VERSION_SUB_KEY>
```

- Registry Value
  - Install— Gibt an, ob die .NET Framework-Version installiert ist.
  - Version— Installierte Version von.NET Framework (Version 4.0 oder niedriger)
  - Release— Ein REG_DWORD Wert, der der installierten .NET Framework-Version (Version 4.5 oder höher) zugeordnet ist
- DLL Files

  Der Scanner extrahiert die Dateiversion aus `mscorlib.dll` und`System.dll`. Wenn diese Dateien vorhanden sind, werden sie der SBOM als verschachtelte Dateikomponenten hinzugefügt. Für .NET Framework Version 4.5 oder höher wird die größte Dateiversion unter den Dateien als Version gemeldet.

- ASP.NET Core Runtime— Sucht `Microsoft.AspNetCore.App.deps.json` in Installationspfaden nach Dateien und extrahiert die Versionszeichenfolge aus dem folgenden Dateipfadmuster.

```
Microsoft.AspNetCore.App/<VERSION>/Microsoft.AspNetCore.App.deps.json
```

- Outlook for Windows— Analysiert die Windows-Registrierung und extrahiert die Version aus dem folgenden Registrierungsschlüssel.

```
HKLM\SOFTWARE\Classes\Local Settings\Software\Microsoft
\Windows\CurrentVersion\AppModel\PackageRepository\Packages
\Microsoft.OutlookForWindows_<VERSION>_<ARCH>__8wekyb3d8bbwe
```

- Microsoft Teams— Analysiert die Windows-Registrierung und extrahiert die Version aus dem folgenden Registrierungsschlüssel.

```
HKLM\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion
\AppModel\PackageRepository\Packages\MSTeams_<VERSION>_<ARCH>__8wekyb3d8bbwee
```

- Microsoft Office 365 / Microsoft 365— Analysiert die Windows-Registrierung und extrahiert die Version aus dem folgenden Registrierungsschlüssel und Wert.
  - Registrierungsschlüssel

    ```
    HKLM\SOFTWARE\Microsoft\Office\ClickToRun\Configuration
    ```

  - Registrierungswert
    - VersionToReport— Microsoft Office-Version
    - ProductReleaseIds— Liste der Produkt-IDs. Dies wird verwendet, um installierte Office-Produkte zu identifizieren. Weitere Informationen zu Produkt-IDs finden Sie product IDsauf der Microsoft Website.
- Microsoft Office Suite— Sammelt die installierten Office-Anwendungen, indem die folgenden ausführbaren Dateien untersucht werden:
  - `EXCEL.EXE` – Microsoft Excel
  - `WINWORD.EXE` – Microsoft Word
  - `POWERPNT.EXE` – Microsoft PowerPoint
  - `OUTLOOK.EXE` – Microsoft Outlook

  Die Versionsnummer in der Windows-Registrierung wird als verbindliche Versionsnummer für jede installierte Office-Anwendung verwendet.

Beispiel **`state.json`** file

Im Folgenden finden Sie ein Beispiel für eine `state.json` Datei, die zum Sammeln der installierten Visual Studio Version verwendet werden kann.

```
{
    "icon": {
        "mimeType": "image/svg+xml",
        "fileName": "product.svg"
    },
    "updateDate": "2025-11-06T05:05:35.6517471Z",
    "installDate": "2025-11-06T05:05:35.6527436Z",
    "enginePath": "C:\\Program Files (x86)\\Microsoft Visual Studio\\Installer\
\resources\\app\\ServiceHub\\Services\\Microsoft.VisualStudio.Setup.Service",
    "installationName": "VisualStudio/17.14.19+36623.8",
    "catalogInfo": {
        "id": "VisualStudio/17.14.19+36623.8",
```

```
        "buildBranch": "d17.14",
        "buildVersion": "17.14.36623.8",
        "localBuild": "build-lab",
        "manifestName": "VisualStudio",
        "manifestType": "installer",
        "productDisplayVersion": "17.14.19",
// truncated
```

Beispiel für eine PURL

Im Folgenden finden Sie jeweils Microsoft Applications ein Beispiel für eine Paket-URL.

```
// PowerShell
Sample PURL: pkg:generic/microsoft/powershell@7.5.3

// NuGet CLI
Sample PURL: pkg:generic/microsoft/nuget@6.14.0

// Visual Studio Code
Sample PURL: pkg:generic/microsoft/visualstudiocode@1.104.2

// Microsoft Edge
Sample PURL: pkg:generic/microsoft/edge@140.0.3485.94

// SharePoint Server
Sample PURL: pkg:generic/microsoft/sharepoint@23.38.219.1

// Microsoft Defender
Sample PURL: pkg:generic/microsoft/defender@4.18.23110.3

// Exchange Server
Sample PURL: pkg:generic/microsoft/exchangeserver@15.2.2562.17

// Visual Studio
Sample PURL: pkg:generic/microsoft/visualstudio@17.14.19

// .NET Core Runtime
Sample PURL: pkg:generic/microsoft/dotnet@8.0.18

// .NET Framework
Sample PURL: pkg:generic/microsoft/dotnet-framework-v4.8.1@4.8.9320.0

// ASP.NET Core Runtime
Sample PURL: pkg:generic/microsoft/aspdotnet@8.0.18
```

```
// Microsoft Teams
Sample PURL: pkg:generic/microsoft/teams@25241.203.3947.4411

// Outlook for Windows
Sample PURL: pkg:generic/microsoft/outlookforwindows@1.2025.916.400

// Microsoft 365 / Office 365
Sample PURL: pkg:generic/microsoft/office@16.0.19127.20264?
product_ids=O365HomePremRetail

// Microsoft Word
Sample PURL: pkg:generic/microsoft/word@16.0.19127.20264

// Microsoft Excel
Sample PURL: pkg:generic/microsoft/excel@16.0.19127.20264

// Microsoft PowerPoint
Sample PURL: pkg:generic/microsoft/powerpoint@16.0.19127.20264

// Microsoft Outlook
Sample PURL: pkg:generic/microsoft/outlook@16.0.19127.20264
```

Microsoft SQL Server Sammlung von Ökosystemen

Unterstützte Anwendungen

- Microsoft SQL Server

Schlüssel-Features

- Liest aus der Windows Registrierung, um installierte Microsoft SQL Server Instanzen zu ermitteln und Versionsinformationen zu extrahieren.
- Erkennt Instanzen in einem zweistufigen Prozess: liest den `InstalledInstances` Wert, löst jeden Instanzpfad aus dem `Instance Names\SQL` Unterschlüssel auf und liest dann die Setup-Informationen aus dem Unterschlüssel jeder Instanz. `Setup`
- Erfasst den Instanznamen, die Basisversion, das Patch-Level, die Edition, das Service Pack (falls vorhanden) und den Registrierungsschlüsselpfad.
- Die Komponentenversion und die PURL verwenden das Patch-Level (vollständige Buildnummer).

Unterstützte Plattformen — Windows

Der Amazon Inspector SBOM Generator liest aus dem folgenden Windows Registrierungsschlüssel, um installierte Instances zu ermitteln:

```
HKLM\SOFTWARE\Microsoft\Microsoft SQL Server
```

Der Scanner liest den `InstalledInstances` Wert, um die Instances aufzuzählen, löst jeden Instance-Pfad aus dem Unterschlüssel auf und liest dann die Setup-Informationen aus dem `Instance Names\SQL` Unterschlüssel jeder Instance. `Setup`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für eine Microsoft SQL Server Instanz.

```
pkg:generic/microsoft/sqlserver@16.0.1000.6
```

MongoDB Sammlung von Ökosystemen

Unterstützte Anwendungen

- MongoDB Server(7.0+, 8.0+)

Schlüssel-Features

- Untersucht mongod Binärdateien, um eingebettete Versionsinformationen zu extrahieren.

> **Note**
>
> Die mongod Binärdatei kann eine Größe von 200 MB überschreiten. Um nach Dateien zu suchenMongoDB, muss die Dateigrößenbeschränkung von Amazon Inspector SBOM Generator so konfiguriert sein, dass Dateien über 200 MB zulässig sind.

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach MongoDB Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/bin/mongod`
- `/usr/local/bin/mongod`

macOS

- `/usr/local/bin/mongod`
- `/opt/homebrew/bin/mongod`

Windows

- `C:\Program Files\MongoDB\Server\bin\mongod.exe`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürMongoDB Server.

```
pkg:generic/mongodb/mongodb-server@8.2.4?platform=linux
```

Mozilla Sammlung von Ökosystemen

Unterstützte Anwendungen

- Firefox
- Firefox ESR
- Thunderbird

Schlüssel-Features

- Analysiert das `application.ini` Manifest, das im Lieferumfang der Mozilla Anwendungen enthalten ist. Der Parser liest nur den `[App]` Abschnitt und stoppt bei der nächsten Abschnittsüberschrift, sodass Schlüssel aus anderen Abschnitten, wie z. B.`[Gecko]`, ignoriert werden.
- Identifiziert das Produkt aus dem `RemotingName` Feld, das in `firefox`firefox-esr, oder aufgelöst wird. `thunderbird` `RemotingName`wird anstelle von verwendet`Name`, da das `Name` Feld sowohl `Firefox` für den Release- als auch für den ESR-Kanal bestimmt ist.

- Extrahiert die Version wortwörtlich aus dem `Version` Feld. Beides `RemotingName` (entspricht einem bekannten Produkt) und `Version` sind erforderlich. Wenn einer der beiden Werte fehlt oder es sich bei dem `RemotingName` Wert nicht um ein bekanntes Mozilla Produkt handelt, wird keine Komponente gemeldet. Davon ausgenommen sind Mozilla Gecko Anwendungen, die ebenfalls eine `application.ini` Datei versenden.
- Entdeckte Anwendungen werden als CycloneDX Komponenten des `application` Typs gemeldet.
- Zeichnet den `SourceRepository` Wert als CycloneDX Eigenschaft und `amazon:inspector:sbom_generator:mozilla:source_repository` nicht als Paket-URL-Qualifier auf, sodass er sich nicht auf die Identifizierung von Sicherheitslücken auswirkt.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator erkennt plattformübergreifend `application.ini` Dateien anhand ihres Namens mit den folgenden Standardpfaden:

Linux

- `/usr/lib/firefox/`, `/usr/lib/firefox-esr/`, `/usr/lib/thunderbird/`
- `/usr/lib64/firefox/`, `/usr/lib64/thunderbird/`
- `/opt/firefox/`, `/opt/thunderbird/`
- `/usr/local/lib/firefox/`, `/usr/local/lib/thunderbird/`

macOS

- `/Applications/Firefox.app/Contents/Resources/`
- `/Applications/Thunderbird.app/Contents/Resources/`

Windows

- `C:\Program Files\Mozilla Firefox\`, `C:\Program Files (x86)\Mozilla Firefox\`
- `C:\Program Files\Mozilla Thunderbird\`, `C:\Program Files (x86)\Mozilla Thunderbird\`

Einschränkungen

- Installationen in nicht standardmäßigen Pfaden stehen nicht auf der standardmäßigen Localhost-Scanliste. Um sie zu scannen, verwenden Sie die `--path` Option, einen Verzeichnisscan durchzuführen.
- Snapund Flatpak Pakete von Firefox und Thunderbird werden nicht unterstützt.
- Installationen von einem Betriebssystem-Paketmanager werden erkannt, wenn sie eine `application.ini` Datei enthalten. Diese Installationen werden auch von den Paketsammlern des Betriebssystems gemeldet.

Beispiel **`application.ini`** file

Das Folgende ist ein Beispiel für den Inhalt einer `application.ini` Datei für eine Firefox ESR Installation.

```
//truncated

[App]
Vendor=Mozilla
Name=Firefox
RemotingName=firefox-esr
Version=140.14.0
SourceRepository=https://hg.mozilla.org/releases/mozilla-esr140

//truncated
```

Das Produkt `firefox-esr` stammt aus dem `RemotingName` Feld, und die Version `140.14.0` stammt aus dem `Version` Feld.

Beispiel für eine PURL

Im Folgenden finden Sie Beispielpaket-URLs für Mozilla Anwendungen.

```
// Firefox
pkg:generic/mozilla/firefox@152.0.6?distro=darwin

// Firefox ESR
pkg:generic/mozilla/firefox-esr@140.14.0?distro=linux

// Thunderbird
```

```
pkg:generic/mozilla/thunderbird@153.0.3?distro=darwin
```

Nginx Sammlung von Ökosystemen

Unterstützte Anwendungen

- Nginx

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt.

Linux

- /usr/sbin/nginx
- /usr/local/nginx
- /usr/local/etc/nginx
- /usr/local/nginx/nginx
- /usr/local/nginx/sbin/nginx
- /etc/nginx/nginx

Windows

- C:\nginx\nginx.exe
- C:\nginx-x.y.z\nginx.exe (x.y.z ist eine beliebige Version)

macOS

- /usr/local/etc/nginx/nginx

Schlüssel-Features

Diese Sammlung untersucht Binärdateien, um eingebettete Versionsinformationen zu extrahieren. Sie sucht nach Versionszeichenfolgen im `.rodata` Abschnitt der ausführbaren Binärdatei (für ELF-Binärdateien aktiviertLinux), `.rdata` Abschnitt (für PE-Binärdateien aktiviertWindows) oder `__ctring` Abschnitt (für MachO Binärdateien).

Beispiel für eine Versionszeichenfolge

Das Folgende ist ein Beispiel für eine Versionszeichenfolge, die in eine Nginx Binärdatei eingebettet ist.

```
nginx version: nginx/1.27.5
```

Die Version `1.27.5` wird extrahiert, um die Nginx Version zu identifizieren.

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für Nginx.

```
Sample PURL: pkg:generic/nginx/nginx@1.27.5
```

Node.JS Runtime-Sammlung

Unterstützte Anwendungen

- Knotenlaufzeit-Binärdatei für Node.JS

Unterstützte Plattformen

Die folgenden Plattformen werden unterstützt. (* ist eine beliebige Version)

Linux

- /usr/local/bin/node
- /usr/bin/node
- //Knoten nodejs/bin
- ~/. nvm/versions/knoten/ */ bin/node
- ~/. local/share/fnm/node-versionen/*/ /node installation/bin
- ~/. asdf/installs/nodejs/ */ bin/node
- ~/. local/share//node/ mise/installs */ bin/node
- ~/.volta/tools/image/node/*/bin/node

Windows

- C:\Program Dateien\nnodejs\node.exe
- C:\Program Dateien (x86)\nnodejs\node.exe
- ~\\ RoamingAppData\ fnm\nOde-Versionen\ *\ Installation\node.exe

macOS

- /opt/homebrew/Cellar/node/*/bin/node

Schlüssel-Features

In dieser Sammlung werden Binärdateien untersucht, um eingebettete Versionsinformationen zu extrahieren. Sie sucht nach Versionszeichenfolgen im `.rodata` Abschnitt der ausführbaren Binärdatei (für ELF-Binärdateien aktiviertLinux), `.rdata` Abschnitt (für PE-Binärdateien aktiviertWindows) oder `__cstring` Abschnitt (für MachO Binärdateien).

Beispiel für eine Versionszeichenfolge

Im Folgenden finden Sie ein Beispiel für eine Versionszeichenfolge, die in eine Node.JS Runtime-Binärdatei eingebettet ist.

```
node.js/v24.11.1
```

Die Version `24.11.1` wird extrahiert, um die Node.JS Laufzeitversion zu identifizieren.

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürNode.JS.

```
Sample PURL: pkg:generic/nodejs/node@24.11.1
```

Ollama Sammlung von Ökosystemen

Unterstützte Anwendungen

- Ollama(lokale LLM-Laufzeit)

Schlüssel-Features

- Untersucht die `ollama` Binärdatei (oder `ollama.exe` aktiviertWindows), um eingebettete Versionsinformationen zu extrahieren.
- Ollamaist eine Go Binärdatei`-ldflags -X`, bei der die Version beim Erstellen über eingefügt wurde. Diese wird im Datenbereich der Binärdatei als Zeichenfolge mit Tilde-Präfix angezeigt (z. B.). `~0.21.0` Bei Homebrew macOS Binärdateien mit -build kann die Version alternativ im Modul build info als vorhanden sein. Go `ollama\tv<version>`

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/local/bin/ollama`
- `/usr/bin/ollama`
- `~/.local/bin/ollama`

macOS

- `/Applications/Ollama.app/Contents/MacOS/ollama`
- `/usr/local/bin/ollama`
- `/opt/homebrew/bin/ollama`

Windows

- `%LOCALAPPDATA%\Programs\Ollama\ollama.exe`
- `%ProgramFiles%\Ollama\ollama.exe`

Beispiel für eine Versionszeichenfolge

Das Folgende ist ein Beispiel für eine Versionszeichenfolge, die in eine Ollama Binärdatei eingebettet ist.

```
~0.21.0
```

Die Version `0.21.0` wird extrahiert, um die Ollama Version zu identifizieren.

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürOllama.

```
Sample PURL: pkg:generic/ollama/ollama@0.21.0?distro=linux
```

Ollama Model Collector Sammlung von Ökosystemen

Unterstützte Anwendungen

- Ollama-CLI

Schlüssel-Features

- Extrahiert lokal zwischengespeicherte AI/ML Modelle, die von installiert wurden Ollama
- Generiert Ollama Paket-URLs

Beispielpfad

Das Folgende ist ein Beispiel für einen zwischengespeicherten Ollama Modellpfad.

```
/usr/share/ollama/.ollama/models/manifests/registry.ollama.ai/library/gemma4/latest
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein Ollama Modell. Der Komponententyp ist`machine-learning-model`.

```
pkg:ollama/gemma4@<hash>
```

Sammlung des OpenSSH-Ökosystems

Unterstützte Anwendungen

- OpenSSH(Version 9)

- OpenSSH(Ausführung 10)

Unterstützte Plattformen Linux/macOS

- `/usr/sbin/sshd`
- `/usr/local/sbin/sshd`

Unterstützte Plattformen Windows

- `C:/Windows/System32/OpenSSH/sshd.exe`
- `C:/Program Files/OpenSSH/sshd.exe`
- `C:/Program Files (x86)/OpenSSH/sshd.exe`
- `C:/OpenSSH/sshd.exe`

Schlüssel-Features

- Untersucht `sshd` Binärdateien, um eingebettete Versionsinformationen zu extrahieren.
- Sucht nach Versionszeichenfolgen im `.rodata` Abschnitt mit ausführbaren Binärdateien (für ELF-Binärdateien aktiviert)Linux, `__cstring` Abschnitt (für Mach-O Binärdateien aktiviertMacOs) oder `.rdata` Abschnitt (für PE-Binärdateien aktiviert). Windows

Beispiel für eine Versionszeichenfolge

Das Folgende ist ein Beispiel für eine Versionszeichenfolge, die in eine OpenSSH Binärdatei eingebettet ist.

```
OpenSSH_9.9p2
```

Die Version `9.9p2` wird extrahiert, um die OpenSSH Version zu identifizieren.

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürOpenSSH.

```
Sample PURL: pkg:generic/openssh/openssh@9.9p2
```

Sammlung des OpenSSL-Ökosystems

Unterstützte Anwendungen

Die Support für OpenSSL-Bibliotheken und Entwicklungspakete ist auf Software beschränkt, die mit offiziellem OpenSSL für 3.0.0-Versionen und höher erstellt wurde. Die Software muss außerdem der semantischen Versionierung folgen. Benutzerdefinierte oder geforkte OpenSSL-Varianten und Versionen unter 3.0.0 werden nicht unterstützt.

Der Amazon Inspector SBOM Generator extrahiert wichtige Paketinformationen für jede installierte OpenSSL-Instance.

Schlüssel-Features

- Extrahiert die Basiszeichenfolge der SEMVER-Version aus der OpenSSL-Header-Datei
- Identifiziert den Verzeichnispfad, der die OpenSSL-Installation enthält

Der Amazon Inspector SBOM Generator sucht nach OpenSSL-Installationen, indem er plattformübergreifend in gängigen Installationspfaden nach der `opensslv.h` Datei sucht.

Beispiel für einen Installationspfad für Linux/Unix

Im Folgenden finden Sie ein Beispiel für einen Installationspfad für Linux/Unix.

```
/usr/local/include/openssl/opensslv.h
/usr/local/ssl/include/openssl/opensslv.h
/usr/local/openssl/include/openssl/opensslv.h
/usr/local/opt/openssl/include/openssl/opensslv.h
/usr/include/openssl/opensslv.h
```

Der Amazon Inspector SBOM Generator extrahiert Versionsinformationen, indem er die `opensslv.h` Datei analysiert und nach den Versionsdefinitionen sucht.

```
# define OPENSSL_VERSION_MAJOR  3
# define OPENSSL_VERSION_MINOR  4
# define OPENSSL_VERSION_PATCH  0
```

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für die OpenSSL-Version.

```
Sample PURL: pkg:generic/openssl/openssl@3.4.0
```

Oracle Database Server-Sammlung

Unterstützte Anwendungen

- Oracle Database

Unterstützte Plattformen Linux

- `/opt/oracle`
- `/u01/app/oracle`

> **Note**
>
> Die Bewertung der Sicherheitslücke gilt nur für Oracle Datenbankserver Version 19 und höher.

Schlüssel-Features

- Untersucht Oracle Binärdateien, um eingebettete Versionsinformationen zu extrahieren.
- Sucht im `.rodata` Abschnitt der ausführbaren Binärdatei nach Versionszeichenfolgen (für ELF-Binärdateien aktiviertLinux).
- Versionsinformationen folgen einem bestimmten Format, das die RDBMS-Versionszeichenfolge enthält.

Beispiel für eine Versionszeichenfolge

Das Folgende ist ein Beispiel für eine Versionszeichenfolge, die in eine Oracle Database Binärdatei eingebettet ist:

```
RDBMS_23.7.0.25.01DBRU_LINUX.X64_240304
```

Die Version `23.7.0.25.01` wird extrahiert, um die Oracle Database Version zu identifizieren.

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürOracle Database.

```
Sample PURL: pkg:generic/oracle/database@23.7.0.25.01
```

PHP Sammlung von Ökosystemen

Unterstützte Anwendungen

- PHP(Version 8.1 und höher)

Schlüssel-Features

- Extrahiert Versionsinformationen aus ausführbaren PHP Binärdateien mithilfe eingebetteter Versionszeichenfolgen.
- Identifiziert den Verzeichnispfad, der die PHP Binärdatei enthält.
- Erkennt automatisch sowohl PHP Standardbinärdateien als auch versionierte Installationen wie `php8.1`php8.2, und. `php8.3`

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach PHP Installationen in gemeinsamen Installationspfaden:

Linux

- `usr/bin/php8.1 through /usr/bin/php8.9`
- `/usr/sbin/php8.1 through /usr/sbin/php8.9`
- `/usr/local/bin/php, /usr/bin/php, /usr/sbin/php`
- `/usr/local/bin/php8.1 through /usr/local/bin/php8.9`(versionierte Binärdateien)

macOS

- `/opt/homebrew/bin/php`
- `/usr/bin/php`
- `/usr/local/bin/php`

Windows

- `C:/php/php.exe`
- `C:/php8.1/php.exe through C:/php8.9/php.exe`(versionierte Verzeichnisse)

Beispiel PHP Versionsextraktion

Der Amazon Inspector SBOM Generator extrahiert Versionsinformationen aus PHP Binärdateien, indem er nach eingebetteten Versionszeichenfolgen nach dem folgenden Muster sucht.

```
X-Powered-By: PHP/8.4.12
```

`8.4.12`wird aus diesem Muster extrahiert, um die Version zu identifizieren. PHP

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein PHP Muster.

```
pkg:generic/php/php@8.4.12
```

Redis Sammlung von Ökosystemen

Unterstützte Anwendungen

- Redis(Version 7.2 und höher)

Schlüssel-Features

- Extrahiert Versionsinformationen aus ausführbaren Redis `redis-server` Binärdateien mithilfe eingebetteter Versionszeichenfolgen.
- Sucht nach Versionszeichenfolgen im `.rodata` Abschnitt mit ausführbaren Binärdateien (für ELF-Binärdateien aktiviertLinux) oder `__cstring` Abschnitt (für Mach-O Binärdateien aktiviert). macOS

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Redis Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/bin/redis-server`

- `/usr/local/bin/redis-server`

macOS

- `/opt/homebrew/bin/redis-server`
- `/usr/local/bin/redis-server`

Beispiel für eine Versionszeichenfolge

Das Folgende ist ein Beispiel für eine Versionszeichenfolge, die in eine Redis Binärdatei eingebettet ist.

```
redis-7.2.6
```

Die Version `7.2.6` wird extrahiert, um die Redis Version zu identifizieren.

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürRedis.

```
pkg:generic/redis/redis@7.2.6
```

Sammlung eigenständiger Ökosysteme für KI-Anwendungen

Der Amazon Inspector SBOM Generator erkennt eigenständige KI-Desktop- und lokale LLM-Anwendungen, die außerhalb eines Paketmanagers installiert sind. Entdeckte Anwendungen werden als CycloneDX Komponenten mit dem `application` Typ und einem `generic` Paket-URL-Typ gemeldet. Die Methode zur Versionsextraktion unterscheidet sich je nach Anwendung, wie in der folgenden Tabelle beschrieben.

Anwendung	Name des Scanners	Quelle der Version	Beispiel für eine PURL
Tabby (TabbyML)	`tabby-installation`	Abschnitt mit binärer Konstante () OTEL `service.version`	`pkg:generic/tabbyml/tabby@<version>`

Anwendung	Name des Scanners	Quelle der Version	Beispiel für eine PURL
Claude Desktop	`claude-desktop-installation`	WindowsPE `ProductVersion` (`StringFileInfo`)	`pkg:generic/anthropic/claude-desktop@<version>`
LM Studio	`lm-studio-installation`	Electron `resources/app/package.json`	`pkg:generic/lmstudio/lm-studio@<version>`
Jan	`jan-installation`	Binärer Anker `<version>` `jan.ai.app` (Tauri)	`pkg:generic/janhq/jan@<version>`
GPT4All	`gpt4all-installation`	Binärer Anker: Die Zeichenfolge mit dem Anwendungsnamen, `GPT4All` gefolgt von einem Null-Byte und `<version>` () Qt	`pkg:generic/nomic-ai/gpt4all@<version>`

Unterstützte Plattformen

Windowsund Linux (Bereich nur für Anwendungen).

> **Note**
>
> Für Installationen mit nicht standardmäßigen Installationspräfixen ist das Argument erforderlich. `--path` Ein Build, dem der erwartete Versionsanker fehlt, liefert keine Komponente.

Windsurfen Sammlung von Ökosystemen

Unterstützte Anwendungen

- Windsurf(vonCodeium)

Schlüssel-Features

- Untersucht Installationen vonWindsurf, einem Electron auf KI basierenden Code-Editor vonCodeium. Extrahiert die Version aus dem `resources/app/package.json` Electron-Manifest der Anwendung.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/share/windsurf/`
- `/usr/bin/windsurf`

Windows

- `%LOCALAPPDATA%\Programs\windsurf\`
- `<SystemDrive>\Program Files\Windsurf\`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürWindsurf.

```
Sample PURL: pkg:generic/codeium/windsurf@1.10.5?distro=linux
```

WordPress Sammlung von Ökosystemen

Unterstützte Komponenten

- WordPress-Core
- WordPressPlug-ins
- WordPressThemen

Schlüssel-Features

- WordPresscore — analysiert die `/wp-includes/version.php` Datei, um den Versionswert aus der Variablen $wp_version zu extrahieren.
- WordPressplugins — analysiert die `/wp-content/plugins/<WordPress Plugin>/readme.txt` Datei oder `/wp-content/plugins/<WordPress Plugin>/readme.md` Datei, um das `Stable` Tag als Versionsstring zu extrahieren.
- WordPressthemes — analysiert die `/wp-content/themes/<WordPress Theme>/style.css` Datei, um die Version aus den Versionsmetadaten zu extrahieren.

Beispiel **`version.php`** file

Das Folgende ist ein Beispiel für eine WordPress `version.php` Core-Datei.

```
// truncated

/**
* The WordPress version string.
*
* Holds the current version number for WordPress core. Used to bust caches
* and to enable development mode for scripts when running from the /src directory.
*
* @global string $wp_version
*/
$wp_version = '6.5.5';

// truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für WordPress Core.

```
Sample PURL: pkg:generic/wordpress/core/wordpress@6.5.5
```

Beispiel **`readme.txt`** file

Das Folgende ist ein Beispiel für eine WordPress `readme.txt` Plugin-Datei.

```
=== Plugin Name ===
Contributors: (this should be a list of wordpress.org userid's)
Donate link: https://example.com/
Tags: tag1, tag2
Requires at least: 4.7
Tested up to: 5.4
Stable tag: 4.3
Requires PHP: 7.0
License: GPLv2 or later
License URI: https://www.gnu.org/licenses/gpl-2.0.html

// truncated
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein WordPress Plugin.

```
Sample PURL: pkg:generic/wordpress/plugin/exclusive-addons-for-elementor@1.0.0
```

Beispiel **`style.css`** file

Das Folgende ist ein Beispiel für eine WordPress `style.css` Theme-Datei.

```
/*
Author: the WordPress team
Author URI: https://wordpress.org
Description: Twenty Twenty-Four is designed to be flexible, versatile and applicable
 to any website. Its collection of templates and patterns tailor to different needs,
 such as presenting a business, blogging and writing or showcasing work. A multitude
 of possibilities open up with just a few adjustments to color and typography. Twenty
 Twenty-Four comes with style variations and full page designs to help speed up the
 site building process, is fully compatible with the site editor, and takes advantage
 of new design tools introduced in WordPress 6.4.
Requires at least: 6.4
Tested up to: 6.5
Requires PHP: 7.0
Version: 1.2
```

```
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Text Domain: twentytwentyfour
Tags: one-column, custom-colors, custom-menu, custom-logo, editor-style, featured-
images, full-site-editing, block-patterns, rtl-language-support, sticky-post,
 threaded-comments, translation-ready, wide-blocks, block-styles, style-variations,
 accessibility-ready, blog, portfolio, news
*/
```

Beispiel PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL für ein WordPress Thema.

```
Sample PURL: pkg:generic/wordpress/theme/avada@1.0.0
```

Zed Sammlung von Ökosystemen

Unterstützte Anwendungen

- Zed(vonZed Industries)

Schlüssel-Features

- Untersucht die eigenständige Zed Binärdatei, die von vertrieben wirdZed Industries.
- Zedist eine Rust Binärdatei ohne Manifest. Die Version wird aus einer konstanten Zeichenfolge extrahiert, die in die Binärdatei eingebettet ist.

Unterstützte Plattformen

Der Amazon Inspector SBOM Generator sucht plattformübergreifend nach Installationen in gemeinsamen Installationspfaden:

Linux

- `/usr/bin/zed`
- `/usr/local/bin/zed`
- `~/.local/bin/zed`

Windows

- `%LOCALAPPDATA%\Zed\`
- `<SystemDrive>\Program Files\Zed\`

Beispiel für eine PURL

Im Folgenden finden Sie ein Beispiel für eine Paket-URL fürZed.

```
Sample PURL: pkg:generic/zed-industries/zed@1.2.6?distro=linux
```

Amazon Inspector SBOM Generator-Zertifikatscans SSL/TLS

In diesem Abschnitt wird beschrieben, wie Sie den Amazon Inspector SBOM Generator zur Inventarisierung SSL/TLS von Zertifikaten verwenden. Er Sbomgen inventarisiert SSL/TLS Zertifikate, indem er nach Zertifikaten an vordefinierten Speicherorten sowie in vom Benutzer bereitgestellten Verzeichnissen sucht. Die Funktion soll es Benutzern ermöglichen, SSL/TLS Zertifikate zu inventarisieren und abgelaufene Zertifikate zu identifizieren. CA-Zertifikate werden auch im Ausgabeinventar angezeigt.

Verwenden Sbomgen Scans von Zertifikaten

Sie können die Erfassung des SSL/TLS Zertifikatinventars mithilfe des `--scanners certificates` Arguments aktivieren. Zertifikatscans können mit jedem der anderen Scanner kombiniert werden. Standardmäßig sind Zertifikatscans nicht aktiviert.

Der Sbomgen durchsucht je nach dem gescannten Artefakt an verschiedenen Orten nach Zertifikaten. In allen Fällen wird Sbomgen versucht, Zertifikate in Dateien mit den folgenden Erweiterungen zu extrahieren.

```
.pem
.crt
.der
.p7b
.p7m
.p7s
.p12
.pfx
```

Der Artefakttyp Localhost

Wenn der Zertifikatsscanner aktiviert ist und der Artefakttyp localhost ist, sucht er Sbomgen rekursiv nach Zertifikaten in`/etc/*/ssl`,, und `/opt/*/ssl/certs` `/usr/local/*/ssl/var/lib/*/certs`, wobei nicht leer ist. * User-provided Verzeichnisse werden rekursiv durchsucht, unabhängig davon, welche Verzeichnisse benannt sind. In der Regel werden CA/system Zertifikate nicht in diesen Pfaden platziert. Diese Zertifikate befinden sich häufig in Ordnern mit dem Namen `pkica-certs`,, oderCA. Sie können auch in den standardmäßigen Localhost-Scanpfaden vorkommen.

Verzeichnis- und Container-Artefakte

Beim Scannen von Verzeichnis- oder Container-Artefakten wird nach Zertifikaten Sbomgen gesucht, die sich irgendwo auf dem Artefakt befinden.

Beispiele für Befehle zum Scannen von Zertifikaten

Im Folgenden finden Sie Beispiele für Befehle zum Scannen von Zertifikaten. Man generiert eine SBOM, die nur Zertifikate in einem lokalen Verzeichnis enthält. Eine andere generiert eine SBOM, die Zertifikate und AlpineDebian, und Rhel Pakete in einem lokalen Verzeichnis enthält. Ein anderer generiert eine SBOM, die Zertifikate enthält, die sich an gängigen Zertifikatsspeicherorten befinden.

```
# generate SBOM only containing certificates in a local directory
./inspector-sbomgen directory --path ./project/ --scanners certificates

# generate SBOM only containing certificates and Alpine, Debian, and Rhel OS packages
 in a local directory
./inspector-sbomgen directory --path ./project/ --scanners certificates,dpkg,alpine-
apk,rhel-rpm

# generate SBOM only containing certificates, taken from common localhost certificate
 locations
./inspector-sbomgen localhost --scanners certificates
```

Beispiel für eine Dateikomponente

Im Folgenden finden Sie zwei Beispielkomponenten für die Suche nach Zertifikaten. Wenn ein Zertifikat abläuft, können Sie sich eine zusätzliche Eigenschaft anzeigen lassen, die das Ablaufdatum angibt.

```
{
      "bom-ref": "comp-2",
      "type": "file",
```

```
      "name": "certificate:expired.pem",
      "properties": [
            {
                "name": "amazon:inspector:sbom_generator:certificate_finding:IN-
CERTIFICATE-001",
                "value": "expired:2015-06-06T11:59:59Z"
            },
            {
                "name": "amazon:inspector:sbom_generator:source_path",
                "value": "/etc/ssl/expired.pem"
            }
      ]
},
{
      "bom-ref": "comp-3",
      "type": "file",
      "name": "certificate:unexpired.pem",
      "properties": [
            {
                "name": "amazon:inspector:sbom_generator:source_path",
                "value": "/etc/ssl/unexpired.pem"
            }
      ]
}
```

Beispiel einer Komponente zur Reaktion auf Sicherheitslücken

Wenn Sie den Amazon Inspector SBOM Generator mit der `--scan-sbom` Markierung ausführen, wird die resultierende SBOM zur Schwachstellensuche an Amazon Inspector gesendet. Im Folgenden finden Sie ein Beispiel für die Suche nach einem Zertifikat für eine Komponente zur Reaktion auf Sicherheitslücken.

```
{
    "advisories": [
        {
            "url": "https://aws.amazon.com/inspector/"
        },
        {
            "url": "https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/
sec_protect_data_transit_encrypt.html"
        }
    ],
    "affects": [
```

```
        {
            "ref": "comp-2"
        }
    ],
    "analysis": {
        "state": "in_triage"
    },
    "bom-ref": "vuln-1",
    "created": "2025-04-17T18:48:20Z",
    "cwes": [
        324,
        298
    ],
    "description": "Expired Certificate: The associated certificate(s) are no longer
 valid. Replace certificate in order to reduce risk.",
    "id": "IN-CERTIFICATE-001",
    "properties": [
        {
            "name": "amazon:inspector:sbom_scanner:priority",
            "value": "standard"
        },
        {
            "name": "amazon:inspector:sbom_scanner:priority_intelligence",
            "value": "unverified"
        }
    ],
    "published": "2025-04-17T18:48:20Z",
    "ratings": [
        {
            "method": "other",
            "severity": "medium",
            "source": {
                "name": "AMAZON_INSPECTOR",
                "url": "https://aws.amazon.com/inspector/"
            }
        }
    ],
    "source": {
        "name": "AMAZON_INSPECTOR",
        "url": "https://aws.amazon.com/inspector/"
    },
    "updated": "2025-04-17T18:48:20Z"
}
```

Sammlung von Amazon Inspector SBOM Generator-Lizenzen

Der Amazon Inspector SBOM Generator hilft dabei, Lizenzinformationen in einer Software-Stückliste (SBOM) nachzuverfolgen. Er sammelt Lizenzinformationen aus unterstützten Paketen für alle Betriebssysteme und Programmiersprachen. Mit standardisierten Lizenzausdrücken in Ihrer generierten SBOM können Sie Ihre Lizenzverpflichtungen nachvollziehen.

Sammeln Sie Lizenzinformationen

Beispielbefehl

Das folgende Beispiel zeigt, wie Lizenzinformationen aus einem Verzeichnis gesammelt werden.

```
./inspector-sbomgen directory --path /path/to/your/directory/ --collect-licenses
```

Beispiel für eine SBOM-Komponente

Das folgende Beispiel zeigt einen Komponenteneintrag in der generierten SBOM.

```
"components": [
    {
      "bom-ref": "comp-2",
      "type": "application",
      "name": "sample-js-pkg",
      "version": "1.2.3",
      "licenses": [
        {
          "expression": "Apache-2.0 AND (MIT OR GPL-2.0-only)"
        }
      ],
      "purl": "pkg:npm/sample-js-pkg@1.2.3",
    }
  ]
```

> **Note**
>
> Wenn ein Paket untergeordnete Komponenten — zum Beispiel DPKG RPM Binärpakete — in einem Quellpaket verschachtelt, meldet jedes untergeordnete Paket seine eigene Lizenz, anstatt die Lizenz des übergeordneten Pakets zu erben oder wegzulassen. Nested-

component Die Lizenzberichterstattung bezieht sich auf Container-, Localhost- und Volume-Scans, die mit ausgeführt werden. `--collect-licenses`

Unterstützte Pakete

Die folgenden Programmiersprachen und Betriebssystempakete werden für die Lizenzerfassung unterstützt.

Target	Paketmanager	Quelle für Lizenzinformationen	Typ
Alma Linux	RPM	• /usr/lib/sysimage/rpm/rpmdb.sqlite • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS
Amazon Linux	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages	BS

Target	Paketmanager	Quelle für Lizenzinformationen	Typ
		• //.db var/lib rpm/Packages	
CentOS	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS
Fedora	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS

Target	Paketmanager	Quelle für Lizenzinformationen	Typ
OpenSUSE	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS
Oracle Linux	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS

Target	Paketmanager	Quelle für Lizenzinformationen	Typ
Photon OS	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS
RHEL	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS

Target	Paketmanager	Quelle für Lizenzinformationen	Typ
Rocky Linux	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS
SLES	RPM	• /usr/lib//rpmdb.sqlite sysimage/rpm • usr/lib/sysimage/rpm//Pakete • /usr/lib/sysimage/rpm/Packages.db • /var/lib/.sqlite rpm/rpmdb • /var/lib/rpm/Packages • //.db var/lib rpm/Packages	BS
Alpine Linux	APK	`/lib/apk/db/installed`	BS
Chainguard	APK	`/lib/apk/db/installed`	BS

Target	Paketmanager	Quelle für Lizenzinformationen	Typ
Debian	DPKG	`/usr/share/doc/*/copyright`	BS
Ubuntu	DPKG	`/usr/share/doc/*/copyright`	BS
Node.js	Javascript	`node_modules/*/package.json`	Programmiersprache
PHP	Composer-Paket	• `composer.lock` • `/vendor/composer/installed.json`	Programmiersprache
Go	Go	`LICENSE`	Programmiersprache
Python	Python/Egg/Wheel	• `.dist-info/METADATA` • `.egg-info` • `.egg-info/PKG-INFO`	Programmiersprache
Ruby	RubyGem	`*.gemspec`	Programmiersprache
Rust	crate	`Cargo.toml`	Programmiersprache

Standardisierung von Lizenzausdrücken

Das SPDX-Lizenzausdrucksformat bietet eine genaue Darstellung der Lizenzbedingungen, die in Open-Source-Software zu finden sind. Der Amazon Inspector SBOM Generator standardisiert alle Lizenzinformationen anhand der in diesem Abschnitt beschriebenen Regeln in SPDX-Lizenzausdrücken. Die Regeln sorgen für Konsistenz und Kompatibilität aller Lizenzinformationen.

Zuordnung von SPDX-Kurzform-IDs

Alle Lizenznamen sind SPDX-Kurzformkennungen zugeordnet. `MIT License` wurde beispielsweise zu `MIT` verkürzt.

Kombination aus mehreren Lizenzen

Sie können mehr als eine Lizenz mit dem `AND` Betreiber kombinieren. Im Folgenden finden Sie einen Beispielbefehl, der zeigt, wie Sie Ihren Befehl formatieren.

```
MIT AND Apache-2.0
```

Benutzerdefiniertes Lizenzpräfix

Benutzerdefinierten Lizenzen wird ein Präfix `LicenseRef` vorangestellt, wie `LicenseRef-CompanyPrivate` z.

Benutzerdefiniertes Ausnahmepräfix

Benutzerdefinierten Ausnahmen wird ein Präfix `AdditionRef-` vorangestellt, wie `AdditionRef-CustomException` z.

Non-license Werte

Werte, die keine Lizenz benennen, werden ignoriert und nicht aufgezeichnet. Dazu gehören das SPDX `NOASSERTION` und die `NONE` Assertions sowie der vollständige Text einer Lizenz.

Was ist eine Paket-URL?

Eine Paket-URL oder PURL ist ein standardisiertes Format, das zur Identifizierung von Softwarepaketen, Komponenten und Bibliotheken in verschiedenen Paketverwaltungssystemen verwendet wird. Das Format erleichtert das Nachverfolgen, Analysieren und Verwalten von Abhängigkeiten in Softwareprojekten, insbesondere bei der Generierung einer Softwareliste (SBOMs).

PURL-Struktur

Die PURL-Struktur ähnelt einer URL und besteht aus mehreren Komponenten:

- `pkg`— Das wörtliche Präfix

- `type`— Der Pakettyp
- `namespace`— Die Gruppierung
- `name`— Der Paketname
- `version`— Die Paketversion
- `qualifiers`— Zusätzliche Schlüssel-Wert-Paare
- `subpath`— Der Dateipfad im Paket

Beispiel für eine PURL

Das Folgende ist ein Beispiel dafür, wie eine PURL aussehen könnte.

```
pkg:<type>/<namespace>/<name>@<version>?<qualifiers>#<subpath>
```

Die generische PURL

Eine generische PURL wird verwendet, um Softwarepakete und Komponenten darzustellen, die nicht in etablierte Paket-Ökosysteme passen, wie npmpypi, oder. maven Sie identifiziert Softwarekomponenten und erfasst Metadaten, die möglicherweise nicht mit bestimmten Paketverwaltungssystemen übereinstimmen. Eine generische PURL ist für eine Vielzahl von Softwareprojekten nützlich, von kompilierten Binärdateien bis hin zu Plattformen wie Apache und. WordPress Sie ermöglicht die Anwendung in einer Vielzahl von Anwendungsfällen, einschließlich kompilierter Binärdateien, Webplattformen und kundenspezifischer Softwareverteilungen.

Wichtige Anwendungsfälle

- Unterstützt kompilierte Binärdateien und ist nützlich für und Go Rust
- Unterstützt Webplattformen wie Apache undWordPress, bei denen ein Paket möglicherweise nicht mit herkömmlichen Paketmanagern verknüpft ist.
- Unterstützt benutzerdefinierte Legacy-Software, indem Organisationen auf intern entwickelte Software oder auf Systeme verweisen können, denen formelle Pakete fehlen.

Beispielformat

Im Folgenden finden Sie ein Beispiel für das generische PURL-Format.

```
pkg:generic/<namespace>/<name>@<version>?<qualifiers>
```

Zusätzliche Beispiele für das generische PURL-Format

Im Folgenden finden Sie weitere Beispiele für das generische PURL-Format.

Kompiliert Go Binary

Das Folgende steht für das `inspector-sbomgen binary` Kompilierte mit einemGo.

```
pkg:generic/inspector-sbomgen?go_toolchain=1.22.5
```

Kompiliert Rust Binary

Das Folgende stellt die `myrustapp` Binärdatei dar, mit der kompiliert wurdeRust.

```
pkg:generic/myrustapp?rust_toolchain=1.71.0
```

Apache project

Das Folgende bezieht sich auf ein HTTP-Projekt unter dem Apache Namespace.

```
pkg:generic/apache/httpd@1.0.0
```

WordPress software

Das Folgende bezieht sich auf eine WordPress Kernsoftware.

```
pkg:generic/wordpress/core/wordpress@6.0.0
```

WordPress Thema

Das Folgende bezieht sich auf ein benutzerdefiniertes WordPress Thema.

```
pkg:generic/wordpress/theme/mytheme@1.0.0
```

WordPress Plugin

Das Folgende bezieht sich auf ein benutzerdefiniertes WordPress Plugin.

```
pkg:generic/wordpress/plugin/myplugin@1.0.0
```

Umgang mit ungelösten oder nicht standardmäßigen Versionsreferenzen im Amazon Inspector SBOM Generator

Der Amazon Inspector SBOM Generator lokalisiert und analysiert unterstützte Artefakte innerhalb eines Systems, indem er Abhängigkeiten direkt aus den Quelldateien identifiziert. Er ist kein Paketmanager und löst keine Versionsbereiche auf, leitet keine Versionen auf der Grundlage dynamischer Verweise ab und verarbeitet auch keine Abfragen in der Registrierung. Es erfasst Abhängigkeiten nur so, wie sie in den Artefakten der Projektquelle definiert sind. In vielen Fällen werden Abhängigkeiten in Paketmanifesten, wie`package.json`, oder `pom.xml`requirements.txt, mit ungelösten oder bereichsbasierten Versionen spezifiziert. Dieses Thema enthält Beispiele dafür, wie diese Abhängigkeiten aussehen könnten.

Empfehlungen

Der Amazon Inspector SBOM Generator extrahiert Abhängigkeiten aus Quellartefakten, löst oder interpretiert jedoch weder Versionsbereiche noch dynamische Verweise. Für genauere Schwachstellenscans und SBOMs empfehlen wir die Verwendung von gelösten, semantischen Versions-IDs in Projektabhängigkeiten.

Java

Denn Maven Projekte können Versionsbereiche verwendenJava, um Abhängigkeiten in der Datei zu definieren. `pom.xml`

```
<dependency>
    <groupId>org.inspector</groupId>
    <artifactId>inspector-api</artifactId>
    <version>(,1.0]</version>
</dependency>
```

Der Bereich gibt an, dass jede Version bis einschließlich 1.0 zulässig ist. Wenn es sich bei einer Version jedoch nicht um eine aufgelöste Version handelt, erfasst der Amazon Inspector SBOM Generator sie nicht, da sie keiner bestimmten Version zugeordnet werden kann.

JavaScript

Denn JavaScript die `package.json` Datei kann Versionsbereiche enthalten, die den folgenden ähneln:

```
"dependencies": {
    "ky": "^1.2.0",
    "registry-auth-token": "^5.0.2",
    "registry-url": "^6.0.1",
    "semver": "^7.6.0"
}
```

Der ^ Operator gibt an, dass jede Version, die größer oder gleich der angegebenen Version ist, zulässig ist. Wenn es sich bei der angegebenen Version jedoch nicht um eine aufgelöste Version handelt, erfasst der Amazon Inspector SBOM Generator sie nicht, da dies zu Fehlalarmen bei der Erkennung von Sicherheitslücken führen kann.

Python

Denn Python die `requirements.txt` Datei kann Einträge mit einem booleschen Ausdruck enthalten.

```
requests>=1.0.0
```

Der >= Operator gibt an, dass jede Version, die größer oder gleich `1.0.0` ist, zulässig ist. Da dieser spezielle Ausdruck keine genaue Version angibt, kann der Amazon Inspector SBOM Generator keine zuverlässige Version für die Schwachstellenanalyse sammeln.

Der Amazon Inspector SBOM Generator unterstützt keine nicht standardmäßigen oder mehrdeutigen Versionskennungen wie Beta, Latest oder Snapshot.

```
pkg:maven/org.example.com/testmaven@1.0.2%20Beta-RC-1_Release
```

> **Note**
>
> Die Verwendung eines nicht standardmäßigen Suffixes, wie Beta-RC-1_Release z. B., entspricht nicht der standardmäßigen semantischen Versionierung und kann nicht auf Sicherheitslücken innerhalb der Amazon Inspector Detection Engine untersucht werden.

Verwenden Zyklon DX Namespaces mit Amazon Inspector

Amazon Inspector stellt Ihnen CycloneDX Namespaces und Eigenschaftsnamen zur Verfügung, die Sie mit SBOMs verwenden können. In diesem Abschnitt werden alle benutzerdefinierten key/value Eigenschaften beschrieben, die Komponenten in SBOMs hinzugefügt werden können. CycloneDX Weitere Informationen finden Sie unter CyclonedX-Eigenschaftstaxonomie auf der Website. GitHub

amazon:inspector:sbom_scanner-Namespace-Taxonomie

Die Amazon Inspector Scan API verwendet den Namespace und hat die folgenden Eigenschaften: `amazon:inspector:sbom_scanner`

Eigenschaft	Beschreibung
`amazon:inspector:sbom_scanner:cisa_kev_date_added`	Gibt an, wann die Sicherheitslücke zum CISA-Katalog bekannter ausgenutzter Sicherheitslücken hinzugefügt wurde.
`amazon:inspector:sbom_scanner:cisa_kev_date_due`	Gibt an, wann die Schwachstelle gemäß dem CISA-Katalog für bekannte Exploited-Schwachstellen behoben werden muss.
`amazon:inspector:sbom_scanner:critical_vulnerabilities`	Gesamtzahl der Sicherheitslücken mit kritischem Schweregrad, die in der SBOM gefunden wurden.
`amazon:inspector:sbom_scanner:exploit_available`	Gibt an, ob ein Exploit für die angegebene Sicherheitslücke verfügbar ist.

Eigenschaft	Beschreibung
`amazon:inspector:sbom_scanner:exploit_last_seen_in_public`	Gibt an, wann ein Exploit für die angegebene Sicherheitslücke zuletzt in der Öffentlichkeit gesehen wurde.
`amazon:inspector:sbom_scanner:fixed_version: component_bom_ref`	Stellt die korrigierte Version der angegebenen Komponente für die angegebene Sicherheitslücke bereit.
`amazon:inspector:sbom_scanner:high_vulnerabilities`	Gesamtzahl der Sicherheitslücken mit hohem Schweregrad, die in der SBOM gefunden wurden.
`amazon:inspector:sbom_scanner:info`	Stellt den Scankontext für eine bestimmte Komponente bereit, zum Beispiel: „Gescannte Komponente: keine Sicherheitslücken gefunden“.
`amazon:inspector:sbom_scanner:is_malicious`	Zeigt an, ob OpenSSF betroffene Komponenten als bösartig identifiziert.
`amazon:inspector:sbom_scanner:low_vulnerabilities`	Gesamtzahl der Sicherheitslücken mit niedrigem Schweregrad, die in der SBOM gefunden wurden.
`amazon:inspector:sbom_scanner:medium_vulnerabilities`	Gesamtzahl der Sicherheitslücken mit mittlerem Schweregrad, die in der SBOM gefunden wurden.
`amazon:inspector:sbom_scanner:path`	Der Pfad zu der Datei, die die Betreff-Paketinformationen enthält.
`amazon:inspector:sbom_scanner:priority`	Die empfohlene Priorität für die Behebung einer bestimmten Sicherheitslücke. Die Werte in absteigender Reihenfolge sind „IMMEDIATE“, „URGENT“, „MODERATE“ und „STANDARD“.

Eigenschaft	Beschreibung
`amazon:inspector:sbom_scanner:priority_intelligence`	Die Qualität der Informationen, die zur Bestimmung der Priorität einer bestimmten Sicherheitslücke verwendet werden. Zu den Werten gehören „VERIFIZIERT“ oder „NICHT VERIFIZIERT“.
`amazon:inspector:sbom_scanner:warning`	Stellt den Kontext dafür bereit, warum eine bestimmte Komponente nicht gescannt wurde, z. B.: „Komponente übersprungen: kein URL angegeben“.

`amazon:inspector:sbom_generator-Namespace-Taxonomie`

Der Amazon Inspector SBOM Generator verwendet den Namespace und hat die folgenden Eigenschaften: `amazon:inspector:sbom_generator`

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:cpu_architecture`	Die CPU-Architektur des Systems, das inventarisiert wird (x86_64).
`amazon:inspector:sbom_generator:ec2:instance_id`	Die Amazon EC2-Instanz-ID.
`amazon:inspector:sbom_generator:ec2:instance_type`	Der Amazon EC2-Instance-Typ
`amazon:inspector:sbom_generator:live_patching_enabled`	Ein boolescher Wert, der angibt, ob Live-Patching auf Amazon EC2 Amazon aktiviert ist. Linux
`amazon:inspector:sbom_generator:live_patched_cves`	Eine Liste der CVEs, die durch Live-Patching auf Amazon EC2 Amazon gepatcht wurden. Linux

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:dockerfile_finding: inspector_finding_id`	Zeigt an, dass ein Amazon Inspector-Fund in einer Komponente mit Prüfungen zusammenhängt. Dockerfile
`amazon:inspector:sbom_generator:image_id`	Der Hash, der zur Container-Image-Konfigurationsdatei gehört (auch bekannt als Image-ID).
`amazon:inspector:sbom_generator:image_arch`	Die Architektur des Container-Images.
`amazon:inspector:sbom_generator:image_author`	Der Autor des Container-Images.
`amazon:inspector:sbom_generator:image:cmd: count`	Ein absolutes Verzeichnis innerhalb des Container-Images, das im Standard-CMD definiert ist, das zum Zeitpunkt der Image-Erstellung konfiguriert wurde.
`amazon:inspector:sbom_generator:image:entrypoint: count`	Ein absolutes Verzeichnis innerhalb des Container-Images, das im Standard-ENTRYPOINT definiert ist, das zum Zeitpunkt der Image-Erstellung konfiguriert wurde.
`amazon:inspector:sbom_generator:image:workdir`	Das WORKDIR-Verzeichnis des Container-Images, das zum Zeitpunkt der Image-Erstellung konfiguriert wurde.
`amazon:inspector:sbom_generator:image:base_layer`	Die Diff-ID der untersten Ebene des Container-Images, die die Basisebene des Images für nachgeschaltete Verbraucher identifiziert.
`amazon:inspector:sbom_generator:image_docker_version`	Die Docker-Version, die zum Erstellen des Container-Images verwendet wurde.
`amazon:inspector:sbom_generator:is_duplicate_package`	Zeigt an, dass das Betreff-Paket von mehr als einem Dateiscanner gefunden wurde.

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:duplicate_purl`	Zeigt die doppelte Paket-PURL an, die von einem anderen Scanner gefunden wurde.
`amazon:inspector:sbom_generator:kernel_name`	Der Kernelname des Systems, das inventarisiert wird.
`amazon:inspector:sbom_generator:kernel_version`	Die Kernelversion des Systems, das inventarisiert wird.
`amazon:inspector:sbom_generator:kernel_component`	Ein boolescher Wert, der angibt, ob ein Themenpaket eine Kernelkomponente ist
`amazon:inspector:sbom_generator:running_kernel`	Ein boolescher Wert, der angibt, ob ein Subjektpaket der laufende Kernel ist
`amazon:inspector:sbom_generator:layer_diff_id`	Der Hash der unkomprimierten Container-Image-Ebene.
`amazon:inspector:sbom_generator:replaced_by`	Der Wert, der das aktuelle Go Modul ersetzt.
`amazon:inspector:sbom_generator:os_hostname`	Der Hostname des Systems, das inventarisiert wird.
`amazon:inspector:sbom_generator:host:process_id: pid:path`	Der Pfad der ausführbaren Datei eines laufenden Prozesses, der während eines Localhost-Scans der Komponente zugeordnet wurde. Wird nur ausgegeben, wenn Sie die `--collect-processes` Option aktivieren.
`amazon:inspector:sbom_generator:host:process_id: pid:started_at`	Die Startzeit (RFC 3339, UTC) eines laufenden Prozesses, der während eines Localhost-Scans der Komponente zugeordnet wurde. Wird nur ausgegeben, wenn Sie die Option aktivieren. `--collect-processes`

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:source_file_scanner`	Der Scanner, der die Datei gefunden hat, die Paketinformationen enthält, zum Beispiel:`/var/lib/dpkg/status` .
`amazon:inspector:sbom_generator:source_package_collector`	Der Collector, der den Paketnamen und die Version aus einer bestimmten Datei extrahiert hat.
`amazon:inspector:sbom_generator:source_path`	Der Pfad zu der Datei, aus der die betreffenden Paketinformationen extrahiert wurden.
`amazon:inspector:sbom_generator:file_size_bytes`	Gibt die Dateigröße eines bestimmten Artefakts an.
`amazon:inspector:sbom_generator:unresolved_version`	Zeigt eine Versionszeichenfolge an, die vom Paketmanager nicht aufgelöst wurde.
`amazon:inspector:sbom_generator:experimental:transitive_dependency`	Zeigt indirekte Abhängigkeiten von einem Paketmanager an.
`amazon:inspector:sbom_generator:subscription:enabled`	Ein boolescher Wert, der angibt, ob ein Abonnement aktiviert ist, z. B. RHEL EUS/E4S oder Pro. Ubuntu
`amazon:inspector:sbom_generator:subscription:name`	Der Name des aktiven Abonnements (z. B. EUS, E4S, Pro).
`amazon:inspector:sbom_generator:subscription:locked_version`	Die Version, die durch das aktive RHEL-Abonnement gesperrt ist (nur RHEL EUS/E4S).
`amazon:inspector:sbom_generator:metadata:host:hostname`	Der Hostname des gescannten Systems.
`amazon:inspector:sbom_generator:metadata:host:kernel_name`	Der Kernelname des Betriebssystems (z. B. Linux, Darwin, Windows_NT).

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:metadata:host:kernel_version`	Die Zeichenfolge mit der Kernelversion des Betriebssystems.
`amazon:inspector:sbom_generator:metadata:host:cpu_architecture`	Die CPU-Architektur des Systems (z. B. x86_64, arm64).
`amazon:inspector:sbom_generator:metadata:host:bootdisk_id`	Eindeutiger Bezeichner der Bootdiskette.
`amazon:inspector:sbom_generator:metadata:host:boot_id`	Eindeutiger Bezeichner für die aktuelle Startsitzung.
`amazon:inspector:sbom_generator:metadata:host:boot_time`	Systemstartzeit im ISO 8601-Format.
`amazon:inspector:sbom_generator:metadata:host:system_id`	Persistenter Systembezeichner (Maschinen-ID unter Linux, MachineGuid unter Windows).
`amazon:inspector:sbom_generator:metadata:host:system_serial`	Hardware-Seriennummer aus der Systemfirmware.
`amazon:inspector:sbom_generator:metadata:host:network_interfaces:` *`name`*`:hardware`	MAC-Adresse der Netzwerkschnittstelle.
`amazon:inspector:sbom_generator:metadata:host:network_interfaces:` *`name`*`:ipv4`	IPv4-Adresse (n), die der Schnittstelle zugewiesen sind.
`amazon:inspector:sbom_generator:metadata:host:network_interfaces:` *`name`*`:ipv6`	Der Schnittstelle zugewiesene IPv6-Adresse (n).

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:metadata:host:sbomgen_tag: key`	Benutzerdefinierte benutzerdefinierte Tags, die über das CLI-Argument --tag übergeben werden.
`amazon:inspector:sbom_generator:metadata:imds:provider`	Der Cloud-Anbieter wurde über IMDS erkannt (aws, azure, gcp).
`amazon:inspector:sbom_generator:metadata:imds:instance_id`	Die Amazon EC2-Instanz-ID oder der Azure-VM-Name.
`amazon:inspector:sbom_generator:metadata:imds:instance_type`	Der Instanztyp (z. B. t3.micro, Standard_D2s_v3).
`amazon:inspector:sbom_generator:metadata:imds:instance_location`	Der region/location der Instanz.
`amazon:inspector:sbom_generator:metadata:imds:instance_partition`	Die Cloud-Partition (aws, aws-cn, aws-us-gov für oder für AWS Azure). AzurePublicCloud
`amazon:inspector:sbom_generator:metadata:imds:account_id`	Die AWS Konto-ID der Amazon EC2-Instance, die aus dem Instanz-Identitätsdokument abgerufen wurde (nur).AWS
`amazon:inspector:sbom_generator:metadata:imds:resource_type`	Der Typ der Cloud-Ressource, die gescannt wird (z. B. aws_ec2_instance).
`amazon:inspector:sbom_generator:metadata:imds:instance_managed_id`	ID der verwalteten Amazon EC2 Systems Manager-Instanz (nur).AWS

Eigenschaft	Beschreibung
`amazon:inspector:sbom_generator:metadata:imds:tenant_id`	Der Microsoft Entra ID-Mandant, zu dem das Abonnement der Instanz gehört (nur Azure, wird aufgefüllt, wenn der VM eine verwaltete Identität zugewiesen wird).
`amazon:inspector:sbom_generator:metadata:imds:resource_group`	Die Azure-Ressourcengruppe, zu der die Instanz gehört (nur Azure).
`amazon:inspector:sbom_generator:metadata:imds:subscription_id`	Die der Instanz zugeordnete Azure-Abonnement-ID (nur Azure).
`amazon:inspector:sbom_generator:metadata:imds:vm_id`	Eindeutige Azure-VM-ID (nur Azure).
`amazon:inspector:sbom_generator:metadata:imds:project_id`	Die Google Cloud-Projekt-ID, zu der die Instanz gehört (nur GCP).
`amazon:inspector:sbom_generator:metadata:imds:numeric_project_id`	Die numerische Google Cloud-Projekt-ID, zu der die Instanz gehört (nur GCP).
`amazon:inspector:sbom_generator:metadata:imds:instance_name`	Der vom GCP-Metadatendienst gemeldete Instanzname (nur GCP).
`amazon:inspector:sbom_generator:metadata:host:open_port: port:protocol`	Zeigt einen offenen Port einer Laufzeitressource an (z. B. EC2)
`amazon:inspector:sbom_generator:hardened_image:vendor`	Der Anbieter eines gehärteten Container-Images

Leitfaden für Plugin-Entwickler

In diesem Handbuch wird erklärt, wie Sie Amazon Inspector SBOM Generator (inspector-sbomgen) mit benutzerdefinierten Lua-Plugins erweitern. Mit Plugins können Sie Unterstützung für neue Paket-Ökosysteme hinzufügen, ohne den Quellcode von sbomgen zu ändern oder neu kompilieren zu müssen.

Den vollständigen Funktionskatalog finden Sie im. the section called “Plugin-API-Referenz” Anleitungen zum Schreiben von Tests finden Sie imthe section called “Leitfaden zum Testen von Plugins”.

-Übersicht

Sbomgen-Plugins sind in Lua geschrieben und folgen einer zweistufigen Pipeline:

- Discovery — scannt die Dateiliste des Artefakts und meldet, welche Dateien für Ihr Ökosystem relevant sind.
- Sammlung — analysiert jede entdeckte Datei und überträgt die Ergebnisse des Pakets in die SBOM.

Event-Modell des Plug-ins

Discovery-Plugins benötigen eine Möglichkeit, Sammlungs-Plugins mitzuteilen, dass Dateien, die Paketmetadaten enthalten, in dem Artefakt im Inventar entdeckt wurden. Um diesen Datenaustausch zu erleichtern, definieren Discovery-Plugins einen Ereignisnamen und geben eine Liste von Dateipfaden zurück. Sammlungs-Plug-ins abonnieren dieses Ereignis und empfangen jeden passenden Dateipfad. Dadurch wird die Dateierkennung vom Parsen entkoppelt — Sie können ein Discovery-Plugin für mehrere Collectors verwenden (Fan-Out-Muster). Jedes Discovery-Plugin muss jedoch einen eindeutigen Eventnamen haben, und jedes Collection-Plugin muss einen eindeutigen Collector-Namen haben. Details dazu finden Sie unter the section called “Kollisionsregeln für Plug-ins”.

Entwickler erkennen dies möglicherweise als das Observer-Entwurfsmuster.

Dieses Design ermöglicht es einem einzigen Discovery-Plugin, mehrere unabhängige Analysen auf performante Weise auszulösen. Ein Discovery-Plugin kann beispielsweise alle Elemente `requirements.txt` in einem Artefakt lokalisieren und dann Folgendes einspeisen:

- Ein Paket-Collector, der jede Zeile nach SBOM-Ergebnissen () analysiert. `name==version`

- Ein Secrets-Collector, der Zeilen mit API-Schlüsseln oder Token, die versehentlich angeheftet wurden, als Versionen kennzeichnet.
- Ein Richtliniensammler, der unverankerte Versionsbezeichner oder Wildcard-Versionsbezeichner meldet.

Jeder Collector wird unabhängig von derselben Dateiliste ausgeführt, ohne dass das Dateisystem des Artefakts erneut durchsucht werden muss. Auf diese Weise können Plugin-Autoren auch neue Sammlungs-Plugins hinzufügen, die bestehende Ereignisse abonnieren, ohne das entsprechende Discovery-Plugin ändern zu müssen.

Schnellstart: Neue Plugins erstellen

Der schnellste Weg, ein neues Plugin zu erstellen, ist die Verwendung des integrierten Scaffolding-Befehls:

```
inspector-sbomgen plugin new
```

Der Befehl fordert zur Eingabe eines Plugin-Namens und eines Projektverzeichnisses auf. Durch Drücken der Eingabetaste wird die in Klammern angegebene Standardeinstellung akzeptiert:

```
Plugin name (identifies the software ecosystem your plugin will inventory, e.g. debian-
dpkg, rhel-rpm, python-pip, cmake) [my-custom-ecosystem]: cmake
Project directory [my-sbomgen-plugins]:
```

Sie können Argumente auch direkt übergeben:

```
inspector-sbomgen plugin new --name cmake --path /tmp/custom-plugins
```

Beginnend mit einem funktionierenden Beispiel

Wenn du mit Plugins experimentieren möchtest, bevor du deine eigene Logik schreibst, erstelle ein neues Plugin mit dem `--with-example` Flag:

```
inspector-sbomgen plugin new --with-example
```

Dadurch wird ein voll funktionsfähiges Plugin-Projekt mit einem Beispiel-Lockfile-Parser, Testdaten und bestandenen Tests generiert. Das Beispiel-Plugin erkennt `example.lock` Dateien, analysiert

`name==version` Einträge und überträgt Pakete in die SBOM. Sie können die Tests sofort ausführen, um das Plugin-System in Aktion zu sehen, und dann den Code so ändern, dass er auf Ihr aktuelles Ökosystem abzielt.

Für fortgeschrittenes Gerüst, das alle optionalen Override-Funktionen (`get_scanner_name`, `get_event_nameget_scanner_groups`, Erkennung mehrerer Ereignisse usw.) beinhaltet, verwende das `--with-overrides` Flag (dazu später mehr):

```
inspector-sbomgen plugin new --with-overrides
```

Vervollständigen Sie Ihr Plugin

Nach dem Aufbau des Gerüsts enthalten die generierten Plugin-Dateien `TODO` Markierungen, die angeben, wo Sie Ihre ökosystemspezifische Logik hinzufügen müssen. Arbeiten Sie sich durch diese Markierungen, um aus dem Gerüst ein funktionierendes Plugin zu machen:

- Ersetze alle `TODO` Markierungen durch deine tatsächlichen Werte
- Aktualisieren Sie das Dateimuster so`discover()`, dass es Ihren Zieldateien entspricht
- Implementieren Sie die Parsing-Logik in jedem `collect()` Paket und rufen Sie `sbomgen.push_package()` es auf

Testen Sie Ihr Plugin

Es gibt zwei Möglichkeiten, deine Plugins zu testen:

1. Built-in Test-Harness — Wird verwendet`inspector-sbomgen plugin test`, um die Plugin-Logik während der Entwicklung zu validieren. Dadurch werden Ihre `init_test.lua` Testdateien ausgeführt, ohne dass ein echtes Artefakt zum Scannen erforderlich ist:

```
inspector-sbomgen plugin test --path ./my-plugins
```

Einzelheiten the section called “Leitfaden zum Testen von Plugins” zum Schreiben von Testdateien und zur Verwendung der API finden Sie im`testing.*`.

2. End-to-end scan — Rufe dein Plugin mit Standardbefehlen von sbomgen auf, um zu überprüfen, ob es gegen echte Artefakte funktioniert. Für diesen Ansatz musst du sowohl ein Artefakt angeben, das die Dateien enthält, auf die dein Plugin abzielt (z. B. ein Verzeichnis mit `requirements.txt` oder ein gleichwertiges Verzeichnis), als auch den Pfad zu deinem Plugin-Verzeichnis:

```
inspector-sbomgen directory \
    --path /path/to/test/dir \
    --plugin-dir ./my-plugins \
    --disable-native-scanners \
    -o sbom.json
```

Das `--disable-native-scanners` Flag stellt sicher, dass nur Ihre Lua-Plugins ausgeführt werden, was das Testen ohne Ausgabe der integrierten (nativen) Scanner erleichtert.

IDE-Setup

Sbomgen bietet Codevervollständigung, Typprüfung und Inline-Dokumentation für die gesamte `sbomgen.*` API in VS Code.

VS-Code mit Lua Language Server

- Installieren Sie die Lua-Erweiterung: sumneko.lua
- Öffne eine beliebige Datei in deinem Plugin-Projekt `.lua`

Das war's! Der `plugin new` Befehl generiert `.vscode/settings.json` und `library/sbomgen.lua` wird automatisch vom Lua Language Server erkannt. Sie erhalten sofort:

- Code-Vervollständigung für alle `sbomgen.*` Funktionen
- Parameterhinweise mit Typen
- Hover-Dokumentation
- Typüberprüfung

Struktur des Plugin-Verzeichnisses

Plug-ins zur Erkennung und Sammlung von Sbomgen müssen der folgenden Verzeichnisstruktur entsprechen:

```
{plugin-dir}/
### discovery/
#   ### {platform}/
#       ### {category}/
#           ### {ecosystem}/
```

```
#               ### init.lua          # REQUIRED entrypoint
### collection/
    ### {platform}/
        ### {category}/
            ### {ecosystem}/
                ### init.lua          # REQUIRED entrypoint
```

Diese Verzeichnisnamen haben eine semantische Bedeutung — sbomgen verwendet sie, um Standardmetadaten für Ihr Plugin abzuleiten, einschließlich des Scannernamens, des Ereignisnamens, der Scannergruppen und der Plattformfilterung. Das reduziert die Menge an Standardtexten, die Entwickler sonst schreiben müssten. Durch die Auswahl der richtigen Werte wird sichergestellt, dass Ihr Plugin ordnungsgemäß in das Scanner-Auswahl- und Ausführungsmodell von sbomgen integriert wird.

In den folgenden Abschnitten wird die Verzeichnisstruktur detaillierter untersucht und Hinweise zur semantischen Bedeutung und zu den Konventionen gegeben.

Plattform

Das Plattformverzeichnis steuert, auf welchen Betriebssystemen Ihr Plugin läuft.

Wert	Wann sollte dies verwendet werden?
`cross-platform`	Das Plugin funktioniert auf jedem Betriebss ystem (die meisten Plugins)
`linux`	Linux-specific Erkennungslogik
`windows`	Windows-specific Erkennungslogik
`macos`	Logik OS-specific zur Mac-Erkennung

Kategorie

Das Kategorienverzeichnis bestimmt die Standard-Scanner-Gruppen, die Ihrem Plugin zugewiesen sind. Dadurch wird gesteuert, ob es standardmäßig ausgeführt wird oder ob eine ausdrückliche Anmeldung erforderlich ist. Erfahren Siethe section called “Auswahl des Scanners”, wie sich Gruppen auf die Ausführung auswirken.

Wert	Standardgruppen	Wann sollte dies verwendet werden?
`proglang`	`programming-language-packages` , `pkg-scanner`	Programmiersprachenpakete (pip, npm, maven usw.)
`os`	`os`, `pkg-scanner`	Betriebssystem-Paketmanager (dpkg, rpm, apk usw.)
`extra-ecosystems`	`extra-ecosystems` , `pkg-scanner`	Anwendungen und Laufzeiten (Nginx, Curl, Wordpress usw.)

Wenn Sie einen Kategorienamen verwenden, der keinem der oben genannten Punkte entspricht, wird der Kategoriename selbst als Gruppe verwendet.

Ökosystem

Ein Name für das spezifische Paket-Ökosystem (z. B.`python-pip`,,`python-poetry`,`debian-dpkg`,`curl`). Namen mit Bindestrich sind eine übliche Konvention, aber keine strikte Anforderung.

Der Scanner-Name und der Collector-Name werden direkt vom Namen des Ökosystem-Verzeichnisses abgeleitet.

Paarung von Ereignisnamen

Discovery- und Collection-Plug-ins im selben Verzeichnispfad werden automatisch gepaart. Beispielsweise verbindet sich ein Discovery-Plug-In at `discovery/cross-platform/proglang/python-pip/` automatisch mit`collection/cross-platform/proglang/python-pip/`.

Sie können dies überschreiben, indem Sie `get_event_name()` und `subscribe_to_event()` in Ihren Plugins definieren.

Discovery-Plugins

Ein Discovery-Plugin benötigt nur die `discover()` Funktion. Alle anderen Funktionen sind optional — Standardwerte werden aus dem Verzeichnispfad abgeleitet.

Die meisten Discovery-Plugins funktionieren, indem sie Dateien lokalisieren, deren Namen oder Pfade ein bestimmtes Ökosystem identifizieren — zum Beispiel `requirements.txt` für Python

pip, `package.json` für npm oder `Cargo.lock` für Rust Cargo. Die `sbomgen.find_files_by_*` Funktionen führen diesen Abgleich außerhalb der Lua-VM durch, wodurch sie deutlich schneller sind als die Iteration der vollständigen Dateiliste in Lua:

```
-- REQUIRED: Scans the artifact and returns a table of file paths.
function discover()
    return sbomgen.find_files_by_name({"requirements.txt"})
end
```

`discover()`muss eine Lua-Tabelle (ein Array) mit Zeichenketten zurückgeben. Wenn keine Dateien gefunden werden, wird eine leere Tabelle `{}` zurückgegeben.

Allgemeine Erkennungsmuster

Ziel	Empfohlene Funktion
Ordnen Sie einen oder mehrere exakte Dateinamen zu	`sbomgen.find_files_by_name({names})`
Ordnen Sie Dateinamen ohne Berücksichtigung der Groß- und Kleinschreibung zu	`sbomgen.find_files_by_name_icase({names})`
Nach Pfadsuffix abgleichen (z. B.) `/pom.properties`	`sbomgen.find_files_by_suffix({suffixes})`
Nach vollständigem Pfadausdruck abgleichen	`sbomgen.find_files_by_path_regex({patterns})`
Glob-style Übereinstimmung mit dem Basisnamen (z. B.) `*.lock`	`sbomgen.glob_find_files(pattern)`

Wenn Ihre Logik eine Nachfilterung erfordert — zum Beispiel Dateien behalten, die einem Suffix entsprechen, aber Build-Output-Verzeichnisse ausschließen — kombinieren Sie einen `find_files_by_*` Aufruf mit einer Lua-Schleife:

```
function discover()
    local found = {}
    for _, f in ipairs(sbomgen.find_files_by_suffix({".conda-meta.json"})) do
```

```
        if not f:match("[/\\]%.cache[/\\]") then
            table.insert(found, f)
        end
    end
    return found
end
```

Vermeiden Sie `sbomgen.get_file_list()` die Erkennung, es sei denn, es passt kein anderer Matcher — es kopiert jeden Pfad in die Lua-VM und kann bei großen Artefakten mehrere Sekunden dauern. Einzelheiten finden Sie in derthe section called “Plugin-API-Referenz”.

Multi-event Entdeckung

Standardmäßig `discover()` werden alle von zurückgegebenen Dateien in einem einzigen Ereignis (von`get_event_name()`) veröffentlicht. Wenn Ihr Scanner verschiedene Dateien an verschiedene Collectors weiterleiten muss, geben Sie stattdessen eine verschlüsselte Tabelle zurück:

```
function discover()
    return {
        EventNameFoundCurl = sbomgen.find_files_by_name({"curl", "curl.exe"}),
        EventNameFoundLibcurl = sbomgen.find_files_by_name({"curlver.h"}),
    }
end
```

Wenn eine Tabelle mit Zeichenkettenschlüsseln `discover()` zurückgegeben wird, wird jeder Schlüssel als separater Ereignisname behandelt und sein Wert (eine Tabelle mit Dateipfaden) wird für dieses Ereignis veröffentlicht. Sammlungs-Plugins abonnieren bestimmte Ereignisse `subscribe_to_event()` wie gewohnt.

Dies ist abwärtskompatibel — die Rückgabe einer sequentiellen Tabelle funktioniert `{"file1", "file2"}` weiterhin im Einzelereignismodus. Die Erkennung erfolgt automatisch: Bei Tabellen mit beliebigen Zeichenkettenschlüsseln handelt es sich um mehrere Ereignisse, bei Tabellen mit nur Integer-Schlüsseln (oder leeren Schlüsseln) handelt es sich um Einzelereignisse.

Bei Verwendung von Mehrfachereignissen `get_event_name()` wird diese Option nicht für die Veröffentlichung verwendet (die Namen der Ereignisse stammen aus den zurückgegebenen Tabellenschlüsseln). Es wird jedoch immer noch beim Laden des Plugins zur Kollisionserkennung aufgerufen, daher sollte es einen eindeutigen Wert zurückgeben oder weggelassen werden, um die Standardeinstellung zu verwenden.

Optionale Erkennungsfunktionen

All diese haben vernünftige Standardwerte, die sich aus dem Verzeichnispfad ergeben. Definieren Sie sie nur, wenn Sie Folgendes überschreiben müssen:

Funktion	Standard	Überschreiben wenn...
`get_scanner_name()`	`{ecosystem}` (z. B. `python-pip`)	Sie möchten einen benutzerdefinierten Scanner-Namen
`get_scanner_description()`	`"Lua discovery plugin: {ecosystem}"`	Sie möchten eine benutzerdefinierte Beschreibung
`get_scanner_groups()`	Abgeleitet aus dem Kategorieverzeichnis	Sie benötigen nicht standardmäßige Gruppen
`get_event_name()`	Abgeleitet vom Verzeichnispfad	Sie benötigen ein benutzerdefiniertes Event-Routing
`get_localhost_scan_paths()`	Keine	Ihr Plugin benötigt bestimmte Pfade, die während der `localhost` Scans gescannt werden
`get_fileless_events()`	Keine	Ein Collector liest eine andere Quelle als das Dateisystem, sodass sein Ereignis auch dann ausgelöst werden muss, wenn nichts `discover()` gefunden wurde

Localhost-Scanpfade

Wenn sbomgen einen `localhost` Scan ausführt, durchsucht es benutzerdefinierte Verzeichnisse sowie alle von Scannern deklarierten Standardpfade. Standardmäßig liefern Lua-Discovery-Plugins keine Pfade, sodass Dateien außerhalb der vom Benutzer angegebenen Verzeichnisse nicht in der Dateiliste erscheinen.

Definiert`get_localhost_scan_paths()`, dass Verzeichnisse oder Dateipfade zurückgegeben werden, die der Localhost Walker enthalten soll:

```
function get_localhost_scan_paths()
    return {
        "/usr/bin",
        "/usr/local/bin",
    }
end
```

Die zurückgegebenen Pfade werden nur bei `localhost` Scans an die Scanliste des Walkers angehängt — sie haben keine Auswirkung auf `containerdirectory`, oder `archive` Scans.

Platform-specific Pfade scannen

Wenn sich die Dateien, die Ihnen wichtig sind, unter Windows, macOS und Linux an unterschiedlichen Speicherorten befinden, verzweigen Sie sich `sbomgen.get_platform()` und geben Sie die entsprechenden Pfade für den Host zurück:

```
function get_localhost_scan_paths()
    local platform = sbomgen.get_platform()

    if platform == sbomgen.platform.WINDOWS then
        local drive = sbomgen.get_system_drive()
        return {
            drive .. "/Program Files/MyApp/myapp.exe",
            drive .. "/Program Files (x86)/MyApp/myapp.exe",
        }
    end

    if platform == sbomgen.platform.DARWIN then
        return {"/Applications/MyApp.app/Contents/MacOS/myapp"}
    end

    -- Linux
    return {
        "/usr/bin/myapp",
        "/usr/local/bin/myapp",
    }
end
```

Verwenden Sie unter Windows, `sbomgen.get_system_drive()` um den Systemlaufwerksbuchstaben aufzulösen (z. B.`"C:"`), anstatt ihn fest zu codieren. Bei Pfaden, die von Umgebungsvariablen wie `LOCALAPPDATA` oder abgeleitet sind`PROGRAMFILES`, iterieren Sie `sbomgen.get_env_vars()` und suchen Sie den Wert anhand des Schlüssels nach. Einzelheiten finden Sie the section called “Plugin-API-Referenz” unter.

Plugins für die Sammlung

Ein Sammlungs-Plugin benötigt nur die `collect()` Funktion. Alle anderen Funktionen sind optional.

`collect(file_path)`wird einmal pro Datei aufgerufen, die vom gekoppelten Discovery-Plugin entdeckt wird. Das typische Muster ist:

- Lesen Sie den Inhalt der Datei mit `sbomgen.read_file()` (für kleine Dateien, die in den Speicher geladen werden) oder `sbomgen.open_file()` (für große Dateien zeilenweise gelesen).
- Analysieren Sie den Inhalt — Zeichenkettenabgleich für einfache Manifeste, `sbomgen.json_decode()` für JSON, für XML oder `sbomgen.xml_decode()` `sbomgen.search_binary()` für kompilierte Binärdateien.
- Veröffentlichen Sie jedes entdeckte Paket, indem Sie es `sbomgen.push_package()` mit den Metadaten des Pakets aufrufen.

```
-- REQUIRED: Called once per discovered file.
-- Parse the file and call sbomgen.push_package() for each package found.
function collect(file_path)
    local content, err = sbomgen.read_file(file_path)
    if err or not content then return end

    for line in content:gmatch("[^\r\n]+") do
        local name, version = line:match("^([%w%-%_%.]+)==(.+)$")
        if name and version then
            sbomgen.push_package({
                name = name,
                version = version,
                purl_type = "pypi",
                component_type = sbomgen.component_types.LIBRARY,
            })
        end
    end
end
```

`collect()`gibt keinen Wert zurück. Jeder `push_package()` Aufruf erfordert`name`,`purl_type`, und`component_type`. Alle unterstützten Felder finden Sie unter. the section called “Plugin-API-Referenz”

Metadaten an Komponenten anhängen

Sbomgen unterstützt zwei Möglichkeiten, Metadaten an eine Paketkomponente anzuhängen: PURL-Qualifizierer und CyclonedX-Eigenschaften. Sie dienen unterschiedlichen Zwecken, und die Wahl zwischen ihnen hat Auswirkungen darauf, wie Amazon Inspector Sicherheitslücken in der resultierenden SBOM identifiziert.

Mechanismus	Wo es erscheint	Verwenden Sie für
`qualifiers`	Innerhalb der Paket-URL (z. B.`pkg:deb/debian/curl@7.88.1?arch=amd64`)	Daten, die Teil der Paketidentität sind
`properties`	Im Array der SBOM `components[].properties`	Beschreibende Metadaten, die nichts daran ändern, wie das Paket identifiziert wird

Empfehlung: Bevorzugen Sie CyclonedX-Eigenschaften (unter Ihrem eigenen Namespace) für benutzerdefinierte Metadaten. Eigenschaften ändern nicht die Identität einer Komponente, sodass sie sich nicht auf die Identifizierung von Sicherheitslücken durch Amazon Inspector auswirken können. Reservieren Sie PURL-Qualifier für Fälle, in denen der PURL-Typ Ihres Ökosystems sie erfordert.

PURL-Qualifikatoren

Einige PURL-Qualifier haben für Amazon Inspector eine semantische Bedeutung und beeinflussen die Identifizierung von Sicherheitslücken. Bei `deb` Komponenten verwendet Inspector beispielsweise Qualifikatoren wie `arch` und, `distro` um den richtigen Schwachstellenfeed auszuwählen, bei `generic` Komponenten für kompilierte Binärdateien Kennzeichner wie `go_toolchain` oder `rust_toolchain` identifizieren die verwendete Toolchain. Das Setzen eines Qualifikators, den Inspector nicht erkennt, oder das Auslassen eines erwarteten Kennzeichners kann dazu führen, dass Sicherheitslücken übersehen oder falsch zugeordnet werden.

Weitere Informationen finden Sie unter Was ist eine Paket-URL? im Amazon Inspector Inspector-Benutzerhandbuch finden Sie Informationen zu den Qualifier-Konventionen, die Inspector pro PURL-Typ erkennt.

Legen Sie die Qualifikatoren über die Tabelle fest auf`qualifiers`: `sbomgen.push_package()`

```
sbomgen.push_package({
    name = "curl",
    version = "7.88.1",
    purl_type = "deb",
    namespace = "debian",
    component_type = sbomgen.component_types.LIBRARY,
    qualifiers = {
        arch = "amd64",
        distro = "debian-12",
    },
})
```

Legen Sie Qualifikatoren nur fest, wenn sie den Erwartungen von Inspector für den PURL-Typ entsprechen. Wenn Sie Metadaten aufzeichnen müssen, die nicht Teil der Paketidentität sind, verwenden Sie stattdessen die CyclonedX-Eigenschaften.

CyclonedX-Eigenschaften

CyclonedX-Eigenschaften sind Anmerkungen zu Schlüsselwerten, die im Array der SBOM erscheinen. `components[].properties` Sie beschreiben eine Komponente, ohne zu beeinflussen, wie sie identifiziert wird, und sind daher die sichere Wahl für Plugin-definierte Metadaten.

Die **`amazon:inspector:*`** Namespaces sind für Amazon Inspector reserviert. Das heißt:

- `amazon:inspector:sbom_generator:*`— reserviert für sbomgen und seine eingebauten Scanner.
- `amazon:inspector:sbom_scanner:*`— reserviert für die Amazon Inspector Scan API.

Plugin-Autoren dürfen keine Schlüssel innerhalb dieser reservierten Namespaces ausgeben. Das Schreiben in sie kann das Verhalten von Inspector beeinträchtigen und kann überschrieben werden. Die vollständige Liste der reservierten Schlüssel finden Sie unter Verwenden von CyclonedX-Namespaces mit Amazon Inspector.

Verwenden Sie Ihren eigenen Namespace (normalerweise Ihre Organisations- oder Plugin-ID), wenn Sie Eigenschaften definieren:

```
sbomgen.push_package({
    name = "requests",
    version = "2.28.1",
    purl_type = "pypi",
    component_type = sbomgen.component_types.LIBRARY,
    properties = {
        ["acme:python:manifest_path"] = file_path,
        ["acme:python:pinned"] = "true",
        ["acme:python:source"] = "requirements.txt",
    },
})
```

Wichtige Benennungsregeln

Eigenschaftsschlüssel werden von sbomgen wie folgt verarbeitet:

- Ein Schlüssel, der einen Doppelpunkt enthält, wird in der SBOM wörtlich verwendet. Fügen Sie immer mindestens einen Doppelpunkt in Ihre Schlüssel ein, damit Sie den Namespace kontrollieren können.
- Einem Schlüssel, der keinen Doppelpunkt enthält, wird automatisch ein Präfix vorangestellt, sodass er im reservierten Inspector-Namespace platziert wird. `amazon:inspector:sbom_generator:` Vermeiden Sie diese Form für benutzerdefinierte Eigenschaften.

```
properties = {
    ["acme:my_plugin:detected_via"] = "lockfile",  -- used as-is (recommended)
    detected_via                    = "lockfile",  -- becomes
 "amazon:inspector:sbom_generator:detected_via" (avoid)
}
```

Die `sbomgen.properties.*` Konstanten existieren, sodass offizielle Scanner konsistente Schlüssel innerhalb des reservierten Namespaces ausgeben. Sie sind keine Erweiterungspunkte für benutzerdefinierte Plugins — verwenden Sie stattdessen Ihren eigenen Namespace.

Eigenschaften und Qualifizierer für untergeordnete Komponenten

Verschachtelt `children` sind unabhängige Komponenten. Jedes untergeordnete Element hat seine eigenen `properties` `qualifiers` Tabellen. Metadaten, die für das übergeordnete Element

festgelegt sind, werden nicht auf Kinder übertragen. Legen Sie Werte explizit für jedes Kind fest, das sie benötigt.

Optionale Sammlungsfunktionen

Funktion	Standard	Überschreiben, wenn...
`get_collector_name()`	`{ecosystem}` (z. B.`python-pip`)	Sie möchten einen benutzerd efinierten Collector-Namen
`get_collector_desc ription()`	leere Zeichenfolge	Sie möchten eine Beschreib ung
`subscribe_to_event()`	Abgeleitet vom Verzeichn ispfad	Sie benötigen ein benutzerd efiniertes Event-Routing
`finalize()`	Keine	Sie gruppieren Ergebnisse in den Dateien eines Ereigniss es, anstatt eines pro Datei auszugeben

Ihre Plugins ausführen

Damit Plugins Paket-Metadaten erzeugen können, muss sbomgen ein Artefakt zum Scannen erhalten, das die Dateien enthält, auf die Ihr Plugin abzielt (z. B. ein Verzeichnis mit `requirements.txtpackage.json`, oder gleichwertige Paketmanifestdateien).

Grundlegende Verwendung

```
inspector-sbomgen <artifact type> <arguments> --plugin-dir /path/to/plugins
```

Beispiel:

```
inspector-sbomgen directory --path /target -o /tmp/sbom.json --plugin-dir /path/to/
plugins
```

Bei deaktivierten nativen Scannern (Modus) Lua-only

```
inspector-sbomgen directory --path /target --plugin-dir /path/to/plugins --disable-
native-scanners -o sbom.json
```

Mit ausführlicher Protokollierung

```
inspector-sbomgen directory --path /target --plugin-dir /path/to/plugins --verbose -o
 sbom.json
```

Verfügbare Scanner auflisten

Dient `list-scanners` zum Anzeigen aller für sbomgen verfügbaren Scanner. Dazu gehören die integrierten nativen Scanner, alle offiziellen Lua-Plugins, die im Lieferumfang von sbomgen enthalten sind, und alle benutzerdefinierten Lua-Plugins, die Sie bereitgestellt haben über: `--plugin-dir`

```
inspector-sbomgen list-scanners --plugin-dir /path/to/plugins
```

```
#################################################################################################
#    SCANNER NAME     # SOURCE #            GROUPS             #         DESCRIPTION
      #
#################################################################################################
# curl                # custom # extra-ecosystems              # Discovers curl version
      #
#                     #        # pkg-scanner                   # header files
 (curlver.h)    #
#################################################################################################
# python-requirements # custom # pkg-scanner                   # Discovers
 requirements*.txt #
#                     #        # programming-language-packages # files for Python pip
      #
#                     #        #                               # packages
      #
#################################################################################################
```

Die Spalte SOURCE zeigt, woher die einzelnen Scanner stammen:

Quelle	Bedeutung
`native`	Built-in Scanner im Lieferumfang von sbomgen

Quelle	Bedeutung
`official`	Mit sbomgen gebündelte Lua-Plugins
`custom`	User-provided Das Lua-Plugin wurde geladen über `--plugin-dir`

Running `list-scanners` without beinhaltet `--plugin-dir` immer noch `native` sowohl `official` Scanner als auch Scanner — diese sind immer verfügbar. Die `--plugin-dir` Flagge fügt Ihre `custom` Scanner zur Liste hinzu.

> Komponenten aus benutzerdefinierten Plugins sind mit dem **`CyclonedX-Bereich gekennzeichnet:`** „optional“
>
> Komponenten, die von vom Benutzer bereitgestellten Lua-Plugins ausgegeben werden, die über geladen wurden, werden in der resultierenden `--plugin-dir` SBOM mit dem CyclonedX-Feld `scope: "optional"` gekennzeichnet. Dies unterscheidet sie von Komponenten, die von integrierten nativen Scannern und offiziellen Lua-Plugins, die im Lieferumfang von sbomgen enthalten sind, erzeugt wurden und `scope` nicht gesetzt sind (Verbraucher verwenden standardmäßig einen fehlenden Bereich gemäß der CyclonedX-Spezifikation). `"required"` Der `scope` Wert gilt für jede Komponente, die das Plugin ausgibt, einschließlich verschachtelter untergeordneter Komponenten, und Sie müssen ihn nicht selbst festlegen. `sbomgen.push_package()`

Um nur Lua-Scanner ohne native Scanner aufzulisten:

```
inspector-sbomgen list-scanners --plugin-dir /path/to/plugins --disable-native-scanners
```

Auswahl des Scanners

Lua-Discovery-Plugins verwenden dasselbe Scanner-Auswahlmodell wie die integrierten nativen Scanner. Standardmäßig führt sbomgen alle Scanner aus, deren Gruppen den Standardscannergruppen für den Artefakttyp entsprechen. Sie können dies mit drei Flags überschreiben:

Führen Sie nur bestimmte Scanner aus

Wird verwendet`--scanners`, um nur die genannten Scanner auszuführen. Alle anderen Scanner sind ausgeschlossen:

```
inspector-sbomgen directory --path /target \
    --plugin-dir /path/to/plugins \
    --scanners python-requirements \
    -o sbom.json
```

Dadurch wird nur der `python-requirements` Scanner ausgeführt. Sie können mehrere durch Kommas getrennte Scannernamen oder einen Scanner-Gruppennamen (z. B.`programming-language-packages`) übergeben, um jeden Scanner zu aktivieren, der zu dieser Gruppe gehört.

Schließen Sie bestimmte Scanner aus

Wird verwendet`--skip-scanners`, um benannte Scanner auszuschließen, während alles andere ausgeführt wird:

```
inspector-sbomgen directory --path /target \
    --plugin-dir /path/to/plugins \
    --skip-scanners python-poetry \
    -o sbom.json
```

Damit werden alle Standardscanner ausgeführt, außer`python-poetry`. Ebenso akzeptiert dieses Flag auch Gruppennamen, sodass durch das Übergeben alle Scanner in dieser Gruppe `--skip-scanners programming-language-packages` deaktiviert werden. `--scanners`

> **Note**
>
> `--scanners`und schließen `--skip-scanners` sich gegenseitig aus. Wenn beide übergeben werden, tritt ein Fehler auf.

Fügen Sie Scanner aus nicht standardmäßigen Gruppen hinzu

Das Standard-Scanner-Set hängt vom Artefakttyp ab, der gescannt wird (siehe die Matrix the section called “Wie wirken sich Gruppen auf die Auswahl aus” unten). Ein Scanner, dessen Gruppen

nicht Teil des Standardsatzes für den Artefakttyp sind, kann nur ausgeführt werden, wenn Sie ihn aktivieren. Verwenden Sie diese `--additional-scanners` Option, um Scanner an den Standardsatz anzuhängen, ohne ihn zu ersetzen:

```
inspector-sbomgen directory --path /target \
    --plugin-dir /path/to/plugins \
    --additional-scanners my-extra-scanner \
    -o sbom.json
```

Dadurch werden alle Standardscanner für den Artefakttyp ausgeführt, plus. `my-extra-scanner` Das Flag akzeptiert eine kommagetrennte Liste von Scannernamen oder Gruppennamen und stapelt sich mit dem Standardsatz, anstatt ihn zu ersetzen. Wird verwendet`list-scanners`, um zu überprüfen, zu welchen Gruppen ein Scanner gehört.

Wie wirken sich Gruppen auf die Auswahl aus

Die `get_scanner_groups()` Funktion in Ihrem Discovery-Plugin bestimmt, zu welchen Gruppen der Scanner gehört. Ob ein Scanner standardmäßig ausgeführt wird, hängt sowohl von seinen Gruppen als auch vom Artefakttyp ab, der gescannt wird. Die folgende Matrix zeigt, welche Gruppen im Standard-Scanner-Set für jeden Artefakttyp enthalten sind:

Group (Gruppieren)	**directory** / **archive**	**container**	**localhost**	**volume**	**binary**
`os`	—	✓	✓	✓	—
`programming-language-packages`	✓	✓	✓	✓	—
`binary`	✓	✓	—	—	✓
`extra-ecosystems`	—	✓	✓	✓	—
`dockerfile`	✓	✓	—	—	—

Group (Gruppieren)	**directory** / **archive**	**container**	**localhost**	**volume**	**binary**
`custom`	✓	✓	✓	✓	✓
`certificate`	—	—	—	—	—
`machine-learning`	—	—	—	—	—
`pkg-scanner`	—	—	—	—	—

Ein ✓ bedeutet, dass jeder Scanner in dieser Gruppe standardmäßig für diesen Artefakttyp ausgeführt wird. A – bedeutet, dass die Gruppe nicht in der Standardeinstellung enthalten ist, sodass ihre Scanner nur ausgeführt werden, wenn sie explizit über `--scanners` oder `--additional-scanners` ausgewählt wurden.

Bemerkenswerte Details:

- **`custom`**ist immer in der Standardeinstellung — benutzerdefinierte Plugins, die über geladen werden, erhalten `--plugin-dir` automatisch die `custom` Gruppe, sodass sie standardmäßig unabhängig vom Artefakttyp ausgeführt werden.
- **`extra-ecosystems`**ist Standard für `container`, `localhost` und `volume` scannt, aber nicht für`directory`,`archive`, oder `binary` scannt. Für diese Typen müssen Sie die Daten `--additional-scanners` (nach Namen oder `extra-ecosystems` Gruppe) angeben, um sie einzubeziehen.
- **`pkg-scanner`**ist informativ — es kennzeichnet einen Scanner als Paketsammler für die Anzeige`list-scanners`, veranlasst aber nicht von sich aus, dass der Scanner ausgeführt wird. Kombinieren Sie ihn mit einer Ausführungsgruppe (z. B.`programming-language-packages`) in`get_scanner_groups()`.

Beispielsweise `{sbomgen.groups.EXTRA_ECOSYSTEMS, sbomgen.groups.PACKAGE_COLLECTOR}` wird ein Plugin, das zurückkehrt, standardmäßig bei Container-, Localhost- und Volume-Scans ausgeführt, erfordert aber `--additional-scanners` (oder`--scanners`) bei Verzeichnis-, Archiv- und Binärscans.

Kollisionsregeln für Plug-ins

Sbomgen erzwingt eindeutige Metadaten für alle geladenen Plugins, um unbemerktes Überschreiben zu verhindern und die Integrität der SBOMs zu gewährleisten. Wenn eine Kollision erkannt wird, wird das spätere Plugin übersprungen und eine Warnung wird protokolliert.

Was wird geprüft

Metadaten	Scope	Bei Kollision
Name des Erkennungsereignisses (`get_event_name`)	Alle Discovery-Plugins	Zweites Plugin wurde übersprungen
Name des Scanners () `get_scanner_name`	Alle Discovery-Plugins	Zweites Plugin wurde übersprungen
Name des Sammlers () `get_collector_name`	Alle Sammlungs-Plugins	Zweites Plugin wurde übersprungen

Was ist erlaubt

Mehrere Sammlungs-Plugins können dieselbe Veranstaltung abonnieren über`subscribe_to_event()`. Dies ist das beabsichtigte Fan-Out-Muster — ein Discovery-Plugin kann mehrere Sammler versorgen, die jeweils unterschiedliche Dinge tun (z. B. extrahiert einer Pakete, ein anderer entdeckt Geheimnisse).

Kollisionen vermeiden

Wenn zwei Plugins denselben Scanner-Namen, Ereignisnamen oder Collector-Namen verwenden, wird das zweite geladene Plug-in übersprungen. Um Kollisionen zu lösen, benennen Sie die widersprüchlichen Metadaten um, indem Sie die entsprechende Override-Funktion in Ihrem Plugin definieren (`get_scanner_name()``get_event_name()`, oder). `get_collector_name()`

Beispiel einer Kollisionswarnung

```
[custom:python-pip] SKIPPED: discovery event name "EventNameFoundPythonRequirements"
is already registered by [official:python-pip]. Each discovery plugin must have a
unique event name. Rename get_event_name() in your plugin to use a unique name.
```

Die Warnung sagt dir, welches Plugin übersprungen wurde, was kollidiert ist, welches Plugin diesen Namen bereits besitzt und welche Funktion geändert werden muss.

Debuggen

Protokollierung in der Konsole

Plugins können mithilfe der folgenden Funktionen Nachrichten an die Konsolenausgabe von sbomgen ausgeben:

Funktion	Stufe	Standardmäßig sichtbar?
`sbomgen.log_debug(message)`	DEBUG	Nein — erfordert `--verbose`
`sbomgen.log_info(message)`	INFO	Ja
`sbomgen.log_warn(message)`	WARN	Ja
`sbomgen.log_error(message)`	ERROR	Ja

Jeder Protokollausgabe eines Plugins wird automatisch die Quelle und der Pfad des Plugins vorangestellt (z. B.`[custom:python-pip]`), sodass Nachrichten von verschiedenen Plugins leicht zu unterscheiden sind. `log_infolog_warn`, und druckt `log_error` immer; druckt `log_debug` nur, wenn sbomgen mit aufgerufen wird. `--verbose`

```
function discover()
    sbomgen.log_info("starting discovery")
    local files = sbomgen.find_files_by_name({"requirements.txt"})
    sbomgen.log_debug(string.format("matched %d files", #files))
    if #files == 0 then
        sbomgen.log_warn("no requirements.txt files found")
    end
    return files
end
```

Breakpoints

Verwenden Sie diese `sbomgen.breakpoint()` Option, um die Ausführung des Plugins anzuhalten und zu blockieren, bis Sie die Eingabetaste drücken. Dies funktioniert wie ein einfacher Debugger — kombinieren Sie ihn mit Protokollanweisungen, um den Status an bestimmten Punkten zu überprüfen.

```
function discover()
    local files = sbomgen.find_files_by_name({"requirements.txt"})

    sbomgen.log_info(string.format("about to inspect %d files", #files))
    sbomgen.breakpoint("before file inspection — press Enter to continue")

    local found = {}
    for _, f in ipairs(files) do
        if not f:match("[/\\]tests[/\\]") then
            table.insert(found, f)
        end
    end

    sbomgen.log_info(string.format("kept %d files after filtering", #found))
    sbomgen.breakpoint("after filtering — press Enter to continue")

    return found
end
```

Die Breakpoint-Nachricht wird auf stderr ausgegeben. Die Ausführung wird angehalten, bis Sie die EINGABETASTE drücken, sodass Sie Zeit haben, die Protokollausgabe zu überprüfen.

Häufige Probleme

Symptom	Ursache	Korrigieren
Das Plugin wurde nicht geladen	`init.lua` fehlt	Stellen Sie sicher, dass der Einstiegspunkt in der richtigen Verzeichnistiefe existiert
„fehlende erforderliche Funktion“	Tippfehler im Funktionsnamen	Prüfen Sie`get_scann er_name` , o`get_scann er_descri ption` ,`get_scann`

Symptom	Ursache	Korrigieren
		`er_groups` ,`discover`,`get_event_name` ,,`get_localhost_scan_paths` ,`get_collector_name` ,`collect`, definiert `subscribe_to_event` sind
Das Sammlungs-Plugin wurde nie aufgerufen	Der Name des Ereignisses stimmt nicht überein	Überprüfen Sie dieselbe `get_event_name()` Zeichenfolge und `subscribe_to_event()` geben Sie sie zurück
Keine Pakete in SBOM	`push_package` nicht aufgerufen oder Pflichtfelder fehlen	Stellen Sie sicher `namepurl_type` , dass, und bei jedem `push_package` Anruf (auch bei Kindern) angegeben `component_type` sind. Verwenden Sie `sbomgen.component_types.*` Konstanten.
Laufzeitfehler im Plugin	Lua-Fehler bei der Ausführung	Überprüfen Sie die Ausgabe von sbomgen auf Warnmeldungen mit den Fehlerdetails
„SKIPPED: Der Name des Erkennungsereignisses... ist bereits registriert“	Ein anderes Plugin verwendet denselben Eventnamen	Benennen `get_event_name()` Sie es in einen eindeutigen Wert um
„SKIPPED: Scannername... ist bereits registriert“	Ein anderes Plugin verwendet denselben Scanner-Namen	Benennen `get_scanner_name()` Sie es in einen eindeutigen Wert um

Symptom	Ursache	Korrigieren
„SKIPPED: Der Name des Kollektors... ist bereits registriert“	Ein anderes Plugin verwendet denselben Collector-Namen	In `get_collector_name()` einen eindeutigen Wert umbenennen

API-Referenz

Der vollständige Funktionskatalog ist in einem Begleitdokument enthalten:

→ the section called “Plugin-API-Referenz”

Die API-Referenz deckt alle `sbomgen.*` Funktionen (Datei I/O, binäre Hilfsprogramme, Paketausgabe, Regex, strukturiertes Parsing, Windows-Registrierung, Protokollierung, Debugging), die in Testdateien verfügbare `testing.*` API, alle integrierten Konstanten (,, `properties` `groups``component_types`,`platform`) und die globalen Plugin-Lebenszykluswerte ab.

Fehlerbehandlung

API-Funktionen, die fehlschlagen können, geben zwei Werte zurück:. `value, err` Bei Erfolg `err` ist`nil`. Bei einem Fehler ist `nil` und `err` ist der erste Wert eine Fehlerzeichenfolge.

```
local content, err = sbomgen.read_file(path)
if err then
    sbomgen.log_error("failed to read " .. path .. ": " .. err)
    return
end
-- content is safe to use here
```

Wenn ein Plugin einen unbehandelten Lua-Fehler auslöst, protokolliert sbomgen eine Warnung und fährt mit der nächsten Datei oder dem nächsten Plugin fort. Andere Plugins sind nicht betroffen.

Sandbox-Einschränkungen

Plugins werden in einer Sandbox-Lua-VM mit eingeschränktem Zugriff auf die Standardbibliothek ausgeführt:

Bibliothek	Verfügbar	Hinweise
`base`	✓	`dofile`,, werden entfernt `loadfile` `loadstring`
`string`	✓	Vollständige Bearbeitung von Zeichenketten
`table`	✓	Vollständige Tabellenm anipulation
`math`	✓	Vollständige Mathe-Bibliothek
`package`	✓	`require()` auf das Plugin-Verzeichnis beschränkt
`io`	✗	Verwenden Sie stattdessen `sbomgen.*` I/O Funktionen
`os`	✗	Aus Sicherheitsgründen gesperrt
`debug`	✗	Blockiert, um VM-Introspektion zu verhindern
`coroutine`	✗	Nicht geladen

Direkter Dateisystemzugriff über `io.open` oder `os.execute` ist nicht verfügbar. Alle Dateioperationen müssen über die `sbomgen` API abgewickelt werden, wodurch ein konsistentes Verhalten bei allen Artefakttypen gewährleistet wird.

Die `sbomgen` Dateifunktionen beschränken Lesevorgänge auf das Artefakt im Inventar: Ein Pfad, der außerhalb der Artefaktwurzel aufgelöst wird (z. B. durch `../` Traversal), wird zurückgewiesen und gibt einen Fehler zurück. Der `localhost` Artefakttyp ist die Ausnahme — er inventarisiert den Host selbst, sodass Lesevorgänge nicht auf eine engere Wurzel beschränkt sind.

`require()`kann Module nur aus dem eigenen Verzeichnisbaum des Plugins laden. Parent-directory Durchquerung, wie sie blockiert `require("../shared")` ist.

Der SBOM-Inhalt wurde nicht bereinigt

Die obige Lesebeschränkung bestimmt, was ein Plugin lesen kann, nicht was es schreibt. Sbomgenüberprüft oder filtert die Daten, die ein Plug-in an die SBOM sendet, nicht und erkennt oder redigiert keine geheimen Daten, Anmeldeinformationen oder andere sensible Werte. Was auch immer ein Plugin in ein Ergebnis einfließen lässt, erscheint in der Ausgabe-SBOM und wird überall dort übertragen, wo diese SBOM veröffentlicht wird. Geben Sie also nur Daten aus, die von dem Artefakt stammen, das Sie inventarisieren möchten.

Code zwischen Plugins teilen

Sie können `require()` es verwenden, um Hilfsmodule aus dem Verzeichnis Ihres Plugins zu laden:

```
my-ecosystem/
### init.lua
### helpers.lua
```

```
-- helpers.lua
local M = {}
function M.parse_version(s)
    return string.match(s, "(%d+%.%d+%.%d+)")
end
return M
```

```
-- init.lua
local helpers = require("helpers")

function subscribe_to_event() return "MyEvent" end

function collect(file_path)
    local content, err = sbomgen.read_file(file_path)
    if err then return end
    local version = helpers.parse_version(content)
    -- ...
end
```

Unterverzeichnisse mit `init.lua` werden ebenfalls unterstützt:

```
my-ecosystem/
### init.lua
### parsers/
```

```
    ### init.lua
```

```
local parsers = require("parsers")
```

`require()`ist auf das Verzeichnis Ihres Plugins beschränkt. Sie können keine Module von anderen Plugins oder Systempfaden laden. Third-party Lua-Bibliotheken (z. B. von LuaRocks) werden nicht unterstützt — nur lokale Hilfsmodule innerhalb des Plugin-Verzeichnisses können geladen werden.

Leitfaden zum Testen von Plugins

Plugin-Autoren schreiben und testen Lua-Plugins vollständig in Lua — es ist keine Go-Toolchain erforderlich. Die Tests befinden sich zusammen mit dem Plugin unter `init_test.lua` und werden über den Befehl ausgeführt. `inspector-sbomgen plugin test`

Allgemeine Informationen zur Plugin-Erstellung finden Sie im. the section called “Leitfaden für Plugin-Entwickler” Den vollständigen Funktionskatalog (einschließlich der `testing.*` API) finden Sie unterthe section called “Plugin-API-Referenz”.

```
-- init_test.lua (next to init.lua)
function test_discovers_curl_version()
    local result = testing.scan_directory("_testdata/include/curl")
    testing.assert_equals(1, #result.findings)
    testing.assert_equals("libcurl", result.findings[1].name)
    testing.assert_equals("8.14.1", result.findings[1].version)
end
```

```
# Run every test under a plugin directory
inspector-sbomgen plugin test --path ./my-plugins

# Verbose output — show each test name and result
inspector-sbomgen plugin test --path ./my-plugins -v
```

Schnellstart

1. Erstellen Sie eine Testdatei

Platziere es `init_test.lua` neben deinem Plugin`init.lua`:

```
my-plugin/
```

```
### discovery/cross-platform/extra-ecosystems/curl/
#   ### init.lua
#   ### init_test.lua
#   ### _testdata/
#       ### include/curl/curlver.h
```

2. Schreiben Sie Testfunktionen

Jede globale Funktion, die mit beginnt, `test_` wird erkannt und ausgeführt:

```
function test_finds_libcurl()
    local result = testing.scan_directory("_testdata/include/curl")
    testing.assert_equals(1, #result.findings)
    testing.assert_equals("libcurl", result.findings[1].name)
end

function test_no_findings_for_empty_dir()
    local result = testing.scan_directory("_testdata/empty")
    testing.assert_equals(0, #result.findings)
end
```

3. Tests ausführen

```
inspector-sbomgen plugin test --path ./my-plugin
```

Verzeichnis-Layout

Struktur des Plugins

Testdateien und Testdaten befinden sich zusammen mit dem Plugin:

```
my-plugins/
### discovery/
#   ### cross-platform/
#       ### extra-ecosystems/
#           ### curl/
#               ### init.lua          # plugin source
#               ### init_test.lua     # test file
#               ### _testdata/        # test fixtures
#                   ### include/curl/curlver.h
#                   ### binaries/unix/curl
```

```
### collection/
#   ### cross-platform/
#       ### extra-ecosystems/
#           ### curl-installation/
#               ### init.lua
#               ### init_test.lua
```

Benennung der Testdateien

- Standard: `init_test.lua` neben den Plugins `init.lua`
- Mehrere Testdateien pro Plugin: Jede passende Datei `*_test.lua` wird erkannt
- Beispiele: `init_test.lua`, `parsing_test.lua`, `discovery_test.lua`.

Testdaten: **`_testdata/`**

Die Testdaten befinden sich `_testdata/` neben dem Plugin. Der führende Unterstrich ist eine Konvention, die Fixtures visuell von der Plugin-Quelle trennt. Der `plugin test` Befehl wird `_testdata/` bei der Suche nach `*_test.lua` Dateien nicht angezeigt, sodass Fixtures niemals mit Testdateien verwechselt werden.

Testdateien verweisen auf Fixtures mit relativen Pfaden:

```
local result = testing.scan_directory("_testdata/include/curl")
```

Pfade werden relativ zu dem Verzeichnis aufgelöst, das die Testdatei enthält.

Das **`Testen.`*** API

Funktionen scannen

Jede Scanfunktion erzeugt ein Artefakt, führt die Discovery → Collection-Pipeline des Plugins aus und gibt Ergebnisse zurück. Der Testautor erstellt Artefakte, Event-Busse oder Register niemals manuell.

```
-- Scan a directory of test fixtures (most common)
local result = testing.scan_directory("_testdata/curl")

-- Alias for scan_directory (archives use the same backend)
local result = testing.scan_archive("_testdata/app.tar.gz")

-- Scan as a localhost artifact
```

```
local result = testing.scan_localhost("_testdata/curl")

-- Scan a compiled binary
local result = testing.scan_binary("_testdata/binaries/curl")

-- Scan a mounted volume
local result = testing.scan_volume("_testdata/volume-root")

-- Scan a container image tarball
local result = testing.scan_container("_testdata/images/alpine.tar")
```

Jede Scanfunktion:

- Erzeugt für jeden Anruf ein neues Artefakt (keine Statuslecks zwischen Aufrufen)
- Lädt nur das Discovery + Collection-Paar des aktuellen Plugins
- Gibt eine Ergebnistabelle zurück

Ergebnistabelle

```
result.findings              -- array of finding tables
result.findings[1].name      -- package name (string)
result.findings[1].version   -- package version (string)
result.findings[1].purl      -- package URL (string)
result.findings[1].component_type -- component type (string)
result.findings[1].properties    -- table<string, string>
result.findings[1].children      -- array of nested finding tables (same shape)
```

Assertionen

```
-- Equality
testing.assert_equals(expected, actual, message?)
testing.assert_not_equals(expected, actual, message?)

-- Truthiness
testing.assert_true(value, message?)
testing.assert_false(value, message?)

-- Nil checks
testing.assert_nil(value, message?)
testing.assert_not_nil(value, message?)
```

```
-- String
testing.assert_contains(haystack, needle, message?)
testing.assert_matches(string, pattern, message?)

-- Tables
testing.assert_length(table, expected_length, message?)

-- Control flow
testing.fail(message)    -- immediately fail the current test
testing.skip(message)    -- skip the current test (not a failure)
```

Standard-Sbomgen. * API

Die vollständige `sbomgen.*` API (Datei I/O, Regex, Systeminformationen, Protokollierung usw.) ist in Testdateien verfügbar, genau wie in Produktions-Plugins. `sbomgen.*`Funktionen, die ein Artefakt benötigen (z. B.`sbomgen.read_file()`), funktionieren jedoch nur innerhalb eines `testing.scan_*` Callbacks — sie sind nicht auf der obersten Ebene einer Testfunktion verfügbar.

Ausführen von Tests

```
# Run all tests under a plugin directory
inspector-sbomgen plugin test --path ./my-plugins

# Filter by regex pattern
inspector-sbomgen plugin test --path ./my-plugins --run curl

# Verbose output (show each test name and result)
inspector-sbomgen plugin test --path ./my-plugins -v

# Stop on the first failing test
inspector-sbomgen plugin test --path ./my-plugins --fail-fast
```

Das `--path` Flag akzeptiert ein Plugin-Stammverzeichnis (enthält `discovery/` and/or `collection/`) oder ein einzelnes Ökosystemverzeichnis (automatisch erkannt). Der Befehl wird mit einem Wert ungleich Null beendet, wenn ein Test fehlschlägt.

Ausgabeformat

Mit `-v` druckt jeder Test eine `=== RUN` Zeile und eine Ergebniszeile (`--- PASS`,`--- FAIL`, oder`--- SKIP`). Ohne werden `-v` nur fehlgeschlagene Tests gedruckt. Am Ende wird eine Übersichtszeile gedruckt:

```
=== RUN   curl/discovery/init_test/test_discovers_libcurl_header
--- PASS: curl/discovery/init_test/test_discovers_libcurl_header (0.04s)
=== RUN   curl/discovery/init_test/test_discovers_curl_binary_unix
--- PASS: curl/discovery/init_test/test_discovers_curl_binary_unix (0.04s)
=== RUN   curl/discovery/init_test/test_no_findings_for_unrelated_files
--- PASS: curl/discovery/init_test/test_no_findings_for_unrelated_files (0.04s)
ok    3 tests passed
```

Plugin-Helfer testen

Die Testdatei wird in dieselbe Lua-VM geladen wie die des `init.lua` Plugins. Im Plugin definierte globale Funktionen können von Tests aus aufgerufen werden. Um Hilfsfunktionen zu testen, stellen Sie sie als globale Funktionen oder in einer Modultabelle zur Verfügung:

```
-- init.lua
M = {}
function M.parse_version(raw)
    return string.match(raw, "(%d+%.%d+%.%d+)")
end

function collect(file_path)
    local ver = M.parse_version(...)
    -- ...
end
```

```
-- init_test.lua
function test_parse_version_extracts_semver()
    testing.assert_equals("1.2.3", M.parse_version("curl/1.2.3"))
end

function test_parse_version_returns_nil_for_garbage()
    testing.assert_nil(M.parse_version("not-a-version"))
end
```

`local`In deklarierte Funktionen `init.lua` sind für die Testdatei nicht sichtbar. Das ist Standard-Lua-Scoping.

Verhalten und Invarianten

Gemeinsam genutzte VM, gemeinsam genutzter Status

Testfunktionen innerhalb einer einzigen Datei teilen sich eine Lua-VM. Eine in einer `test_*` Funktion festgelegte globale Variable ist für nachfolgende Funktionen sichtbar. Wenn zwei Funktionen dieselbe globale Definition definieren, überschreibt die zweite die erste. Jeder Test sollte eigenständig sein und nicht vom Status anderer Tests abhängen.

Non-deterministic Reihenfolge der Ausführung

Testfunktionen werden durch Iteration der globalen Tabelle von Lua erkannt, die eine Hash-basierte Reihenfolge verwendet. Es kann nicht garantiert werden, dass Tests in der Reihenfolge ausgeführt werden, in der sie definiert sind. Schreiben Sie keine Tests, die von der Ausführungsreihenfolge abhängen.

Neues Artefakt pro Scanaufruf

Jeder Aufruf `testing.scan_directory()` (oder jede Scanfunktion) erzeugt ein völlig neues Artefakt. Es gibt keinen Status zwischen Scanaufrufen innerhalb eines Tests oder zwischen Tests.

Das Plugin wird geladen

Plugins werden einmal pro Testlauf geladen, nicht einmal pro Testdatei. Der Test-Runner lädt alle Plugins aus dem bereitgestellten Dateisystem und ordnet dann jede Testdatei der entsprechenden Plugin-VM nach Phase, Plattform, Kategorie und Ökosystem zu.

Verhalten der Assertion

Wenn eine Assertion fehlschlägt, wird der Fehler aufgezeichnet, aber die Testfunktion wird weiterhin ausgeführt — nachfolgende Assertionen und Anweisungen werden weiterhin ausgeführt. Wenn mehr als eine Assertion im selben Test fehlschlägt, ist der jüngste Fehler die in der Zusammenfassung gemeldete Meldung. Frühere Fehler werden überschrieben. Um einen Test beim ersten Fehler zu beenden, kehren Sie nach der fehlgeschlagenen Assertion von der Funktion zurück (oder verwenden Sie ihn `testing.fail()` innerhalb einer Bedingung).

Einschränkungen

- **`local`**Funktionen sind nicht testbar. Nur globale Funktionen von `init.lua` sind für die Testdatei sichtbar. Stellen Sie Helfer über eine Modultabelle zur Verfügung, wenn sie getestet werden müssen.
- Es werden keine I/O externen Scan-Aufrufe für **`sbomgen.*`** Dateien angezeigt. Funktionen wie `sbomgen.read_file()` benötigen einen Artefaktkontext, der nur innerhalb von `testing.scan_*` Aufrufen existiert.
- Keine Lifecycle-Hooks. Es gibt kein `before_each`,`after_each`,`setup`, oder`teardown`. Jede Testfunktion verwaltet ihren eigenen Status.
- Kein Test-Timeout. Eine Testfunktion, die für immer wiederholt wird, lässt den Runner hängen.
- Keine Berichterstattung. Es gibt keine Möglichkeit zu messen, welche Linien trainiert `init.lua` wurden.
- Es gibt keine Benchmarks. Das Testframework unterstützt keine Leistungsbenchmarks.

Verantwortlichkeiten des Entwicklers

Beim Schreiben eines neuen Lua-Plugins

- Erstelle `init_test.lua` neben deinem `init.lua`
- Erstellen Sie `_testdata/` mit minimalen Vorrichtungen, die die Logik Ihres Plugins trainieren
- Schreiben Sie `test_*` Funktionen, die Folgendes abdecken: erfolgreiche Erkennung, Versionsextraktion, Grenzfälle und Szenarien ohne Übereinstimmung
- Führen Sie Tests vor dem Absenden lokal aus

Richtlinien für Testdaten

- Halten Sie die Fixtures auf ein Minimum — verwenden Sie die kleinste Datei, die das Verhalten trainiert
- Vermeiden Sie es, große Binärdateien zu speichern, `_testdata/` wenn eine kleine Textfixierung ausreichen würde
- Jedes Plugin `_testdata/` sollte eigenständig sein — keine Verweise auf Dateien außerhalb des Plugin-Verzeichnisses. Zur Laufzeit sind `sbomgen.*` Lesevorgänge auf das Artefakt im

Inventar beschränkt, sodass ein Plugin, das von Dateien an anderer Stelle auf dem Host abhängt, außerhalb `localhost` von Scans nicht funktioniert.

- Verwenden Sie repräsentative, unempfindliche Geräte. Sbomgenredigiert keine SBOM-Inhalte, also geben Sie niemals echte Geheimnisse oder Zugangsdaten an. `_testdata/`

Plugin-API-Referenz

Vollständige API-Referenz für Inspector-Sbomgen-Lua-Plugins. Eine Anleitung zum Schreiben von Plugins finden Sie unter. the section called “Leitfaden für Plugin-Entwickler” Informationen zum Testen finden Sie unterthe section called “Leitfaden zum Testen von Plugins”.

-Übersicht

Auf alle von der Laufzeit bereitgestellten Funktionen wird über die globale `sbomgen` Tabelle zugegriffen (Datei I/O, Regex, Protokollierung, Konstanten usw.). Darüber hinaus definiert jedes Plugin einen kleinen Satz globaler Funktionen der obersten Ebene (`discover`,,, usw.) `collect` `get_scanner_name``subscribe_to_event`, die sbomgen an definierten Punkten im Plugin-Lebenszyklus aufruft. Diese sind dokumentiert in. the section called “Globaler Lebenszyklus von Plugins”

In `*_test.lua` Dateien stellt sbomgen zusätzlich eine `testing` globale Option zur Verfügung, die es Testautoren ermöglicht, die Discovery→Collection-Pipeline zu steuern und Aussagen zu treffen. Siehe the section called “API testen”.

Sandbox-Einschränkungen

Plugins werden in einer Sandbox-Lua-VM mit eingeschränktem Zugriff auf die Standardbibliothek ausgeführt. Die folgenden Lua-Standardbibliotheksmodule sind verfügbar:

Modul	Hinweise
`base`	Kernfunktionen (`printtype`,`tostring`,`tonumber`,`pairs`,`ipairs`,`pca` `rawgetrawset`,, usw.). `dofile`,`loadfile`, und `loadstring` werden entfernt.

Modul	Hinweise
`string`	Vollständige Bearbeitung von Zeichenketten (`string.match string.find string.format ,string.gsub` ,, usw.)
`table`	Vollständige Tabellenmanipulation (`table.insert table.remove ,table.sort ,table.concat` , usw.)
`math`	Vollständige Mathe-Bibliothek (`math.floor math.maxmath.min`,, usw.)
`package`	`require()` ist verfügbar, aber auf Module innerhalb des eigenen Verzeichnisbaums des Plugins beschränkt. Parent-directory traversal (`require("../shared")`) ist blockiert. `package.cpath` und `package.path` sind gelöscht.

Die folgenden Standardbibliotheksmodule sind aus Sicherheits- und Stabilitätsgründen ausdrücklich nicht zulässig:

Modul	Grund
`io`	Der direkte Dateisystemzugriff ist blockiert. Alle Dateioperationen müssen `sbomgen.*` Funktionen durchlaufen, die über die Artefaktschnittstelle weiterleiten, um ein einheitliches Verhalten bei allen Artefakttypen (Verzeichnis, Container, Volume usw.) zu gewährleisten und Lesevorgänge auf das Artefakt im Inventar zu beschränken (siehe). the section called “Grenze für den Dateizugriff”
`os`	System-level Operationen (`os.execute` ,, `os.remove os.rename` , usw.) werden

Modul	Grund
	blockiertos.getenv , um zu verhindern, dass Plugins das Hostsystem verändern.
debug	Die Debug-Bibliothek ist blockiert, um eine Überprüfung oder Änderung der internen Funktionen der Lua-VM zu verhindern.
coroutine	Coroutinen werden nicht geladen.

Diese Module befinden sich nicht in der Zulassungsliste der VM und sie können von Plugins nicht aufgerufen werden.

> **Note**
>
> Wichtig: Alle Dateien I/O müssen sbomgen.* Funktionen (z. B., sbomgen.read_filesbomgen.open_file,sbomgen.get_file_list) durchlaufen. Die Verwendung von io.open oder irgendein direkter Dateisystemzugriff führt zu einem Laufzeitfehler. Die sbomgen API stellt sicher, dass Plugins mit der Artefakt-Abstraktionsebene interagieren, was für ein einheitliches Verhalten sorgt, unabhängig davon, ob ein Verzeichnis, ein Container-Image, ein Archiv oder ein Volume gescannt wird.

Globaler Lebenszyklus von Plugins

Ein Plugin ist eine Lua-Datei mit dem Nameninit.lua, die bestimmte globale Funktionen der obersten Ebene definiert. Diese Globals sind nicht in der sbomgen Tabelle — es sind Funktionen, die das Plugin für den Aufruf durch sbomgen definiert. Der Satz gültiger Globalwerte unterscheidet sich zwischen Discovery-Plugins und Sammlungs-Plugins. Wenn das Plugin sie auslässt, wird für jede der unten aufgeführten Funktionen die in der Tabelle angegebene Standardeinstellung verwendet.

Discovery-Plugins

Funktion	Arity	Erforderlich	Standard (wenn weggelassen)	Beschreibung
`discover()`	0	Ja	—	Gibt die Dateien zurück, die dieses Plugin gefunden hat. Gibt eine sequentielle Tabelle mit Pfadzeich enfolgen zurück (Einzeler eignismod us) oder eine Tabelle, die durch Zeichenketten mit Ereignisn amen gekennzei chnet ist, deren Werte Pfadtabellen sind (Mehrerei gnismodus).
`get_event_name()`	0	Nein	`"lua:{platform}/{category}/{ecosystem}"`	Gibt den Namen des Ereigniss es zurück, unter dem Dateien veröffent licht werden. Muss für alle Discovery-Plug-

Funktion	Arity	Erforderlich	Standard (wenn weggelassen)	Beschreibung
				ins eindeutig sein.
`get_scanner_name()`	0	Nein	Name des Ökosystem-Verzeichnisses	Gibt den Anzeigenamen des Scanners zurück. Muss für alle Discovery-Plug-ins eindeutig sein.
`get_scanner_description()`	0	Nein	`"Lua discovery plugin: {ecosystem}"`	Gibt eine für Menschen lesbare Beschreibung zurück.
`get_scanner_groups()`	0	Nein	Abgeleitet aus dem Kategorieverzeichnis (siehe Entwicklerhandbuch)	Gibt eine Tabelle mit Zeichenketten für Scanner-Gruppen zurück. Verwenden Sie `sbomgen.groups.*` Konstanten.

Funktion	Arity	Erforderlich	Standard (wenn weggelassen)	Beschreibung
`get_localhost_scan_paths()`	0	Nein	—	Gibt eine Tabelle mit file/directory Pfaden zurück, die beim Scannen eines Localhost-Artefakts berücksichtigt werden sollen. Wird nur für Scans verwendet `localhost` .
`get_fileless_events()`	0	Nein	—	Gibt eine Tabelle mit Ereignisnamen zurück, die veröffentlicht werden sollen, auch wenn keine Dateien für sie `discover()` gefunden wurden. Siehe the section called “Sammeln ohne gefundene Datei”.

Plugins für Sammlungen

Funktion	Arity	Erforderlich	Standard (wenn weggelassen)	Beschreibung
`collect(file_path)`	1	Ja	—	Wird einmal pro Datei aufgerufen, die für das abonnierte Ereignis veröffentlicht wurde. Analysiert die Datei und sendet die Ergebnisse über. `sbomgen.push_package()` Gibt nichts zurück.
`subscribe_to_event()`	0	Nein	`"lua:{platform}/{category}/{ecosystem}"`	Gibt den Namen des Ereignisses zurück, den dieser Collector abonniert hat. Sollte mit den entsprechenden Discovery-Plugins `get_event_name()` übereinstimmen.
`get_collector_name()`	0	Nein	Name des Ökosystem-Verzeichnisses	Gibt den Anzeigenamen des Collectors zurück. Muss für alle Sammlungs-

Funktion	Arity	Erforderlich	Standard (wenn weggelassen)	Beschreibung
				Plugins eindeutig sein.
`get_collector_description()`	0	Nein	`""`(leer)	Gibt eine für Menschen lesbare Beschreibung zurück.
`finalize()`	0	Nein	—	Wird einmal aufgerufen, nachdem die letzte Datei eines Ereignisses durchlaufen wurde. `collect()` Verwenden Sie es, um Ergebnisse in mehreren Dateien gruppiert auszugeben. Siehe the section called “Gruppieren von Ergebnissen nach Dateien”.

Datei I/O

Alle Dateioperationen müssen über die `sbomgen.*` API abgewickelt werden. Direkter Dateisystemzugriff über die `io` Bibliothek von Lua ist nicht verfügbar (siehethe section called “Sandbox-Einschränkungen”). Die `sbomgen` I/O Dateifunktionen werden über die Artifact-Schnittstelle

weitergeleitet und stellen so sicher, dass Ihr Plugin identisch funktioniert, egal ob es ein Verzeichnis auf der Festplatte, ein Container-Image, ein komprimiertes Archiv oder ein gemountetes Volume scannt.

Grenze für den Dateizugriff

Die `sbomgen.*` Dateifunktionen beschränken Lesevorgänge auf das Artefakt im Inventar. Ein Pfad, der außerhalb der Artefaktwurzel aufgelöst wird — zum Beispiel durch `../` Traversal — wird zurückgewiesen, und der Aufruf gibt einen Fehler zurück, anstatt das Host-Dateisystem zu lesen. Dies gilt für,`read_file`,`open_file`, und die binary/hash Helfer `read_dirfile_stat`, die einen Pfad verwenden.

Die Ausnahme ist der `localhost` Artefakttyp, der den Host selbst inventarisiert; dort ist das Host-Dateisystem das Artefakt, sodass Lesevorgänge nicht auf eine engere Wurzel beschränkt sind.

Diese Grenze gilt nur für Lesevorgänge von Dateien. Sie schränkt nicht ein, was ein Plugin in die SBOM schreibt — siehe. the section called “Der SBOM-Inhalt wird nicht bereinigt”

`sbomgen.get_file_list ()`

Gibt alle Dateipfade im Artefakt als Tabelle mit Zeichenketten zurück.

- Gibt zurück: `{string, ...}` — Tabelle mit absoluten Dateipfadzeichenfolgen
- Leistung: Diese Funktion kopiert jeden Dateipfad im Artefakt als Lua-Zeichenfolge in die Lua-VM. Bei großen Artefakten (z. B. einem Localhost-Scan mit mehr als 300.000 Dateien) dauert dies allein mehrere Sekunden. Das Iterieren der zurückgegebenen Tabelle in Lua mit `string.match()` fügt weiteren Aufwand hinzu — ein vollständiger Scan kann mehr als 15 Sekunden dauern. Je mehr Dateien das Artefakt enthält, desto langsamer wird dein Plugin sein.

> **Note**
>
> Bevorzugen Sie diese gezielten Alternativen, wann immer dies möglich ist:
>
> | Funktion | Verwenden Sie, wenn... |
> | --- | --- |
> | `sbomgen.find_files_by_name()` | Sie kennen die genauen Dateiname n, denen Sie entsprechen müssen (z. B.`"requirements.txt"` ,`"curl"`) |

Funktion	Verwenden Sie, wenn...
`sbomgen.find_files_by_name_icase()`	Wie oben, aber unabhängig von Groß- und Kleinschreibung
`sbomgen.find_files_by_suffix()`	Sie müssen Pfadsuffixe abgleichen (z. B.,,) `"/pom.properties"` `"curlver.h"`
`sbomgen.find_files_by_path_regex()`	Sie benötigen einen vollständigen Pfadabgleich mit Regex
`sbomgen.glob_find_files()`	Sie benötigen einen Basisnamenabgleich im Glob-Stil

Diese Funktionen führen den Abgleich außerhalb der Lua-VM durch und geben nur die übereinstimmenden Pfade zurück. Selbst bei 300 Artefakten dauert der Abgleich weniger als 1 Millisekunde. K-file Verwenden Sie diese `get_file_list()` Option nur, wenn Ihre Abgleichslogik mit keinem der oben genannten Werte ausgedrückt werden kann.
Die `glob_find_files` Hilfsprogramme `find_files_by_*` und überspringen Symlinks und geben nur konkrete Dateien zurück, sodass ein Symlink-Alias und sein Ziel nicht beide inventarisiert werden. `get_file_list()`gibt jeden Eintrag zurück, einschließlich Symlinks.

```
-- AVOID in discovery plugins when possible:
local files = sbomgen.get_file_list()
for _, f in ipairs(files) do
    if string.match(f, "pattern$") then ... end
end

-- PREFER:
local matches = sbomgen.find_files_by_name({"target-file.txt"})
```

`sbomgen.read_file (Pfad)`

Liest den gesamten Inhalt einer Datei und gibt ihn als Zeichenfolge zurück.

- Gibt zurück: `string, err`
- Bei einem Fehler: `nil, error_string`

```
local content, err = sbomgen.read_file("/app/package.json")
if err then
    sbomgen.log_error("read failed: " .. err)
    return
end
```

sbomgen.open_file (Pfad)

Öffnet eine Datei zum Streamen von Lesevorgängen. Gibt ein FileHandle Objekt zurück. Verwenden Sie dies für große Dateien, bei denen das Laden des gesamten Inhalts in den Speicher nicht praktikabel ist.

- Gibt zurück: `FileHandle, err`

```
local fh, err = sbomgen.open_file(path)
if err then return end
local line = fh:read_line()
while line do
    -- process line
    line = fh:read_line()
end
fh:close()
```

sbomgen.glob_find_files (Muster)

Gibt Dateien zurück, die einem Go-Glob-Muster entsprechen. `filepath.Match` Das Muster wird mit dem Basisdateinamen abgeglichen. Symlinks werden übersprungen; es werden nur konkrete Dateien zurückgegeben.

- Gibt zurück: `{string, ...}, err`

```
local files, err = sbomgen.glob_find_files("*.txt")
```

Verwenden Sie `sbomgen.get_file_list()` with `string.match` für den vollständigen Pfadmusterabgleich.

sbomgen.find_files_by_name (Namen)

Gibt Dateien zurück, deren Basisname (letzte Pfadkomponente) exakt einem der angegebenen Namen entspricht. Die Iteration und der Vergleich erfolgen in Go, was dies deutlich schneller macht als die Iteration `sbomgen.get_file_list()` in Lua.

- Parameter: `names` — Tabelle mit Zeichenketten (passende Basisnamen)
- Gibt zurück: `{string, ...}` — übereinstimmende Dateipfade, ausgenommen symbolische Links (kein Fehlertupel)

```
local curl_bins = sbomgen.find_files_by_name({"curl", "curl.exe"})
local headers = sbomgen.find_files_by_name({"curlver.h"})
```

sbomgen.find_files_by_name_icase (Namen)

Gibt Dateien zurück, deren Basisname mit einem der angegebenen Namen übereinstimmt, wobei Groß- und Kleinschreibung ignoriert wird. Stimmt beispielsweise `"version"` mit `VERSIONVersion`, und überein. `version` Der `find_files_by_name` Abgleich findet sozusagen außerhalb der Lua-VM statt.

- Parameter: `names` — Tabelle mit Zeichenketten (abzugleichende Basisnamen, Groß- und Kleinschreibung wird nicht beachtet)
- Gibt zurück: `{string, ...}` — übereinstimmende Dateipfade, ohne symbolische Links (kein Fehlertupel)

```
local version_files = sbomgen.find_files_by_name_icase({"version"})
local war_files = sbomgen.find_files_by_name_icase({"jenkins.war"})
```

sbomgen.find_files_by_suffix (Suffixe)

Gibt Dateien zurück, deren vollständiger Pfad (durch Schrägstrich normalisiert) mit einem der angegebenen Suffixe endet. Der Abgleich findet sozusagen außerhalb der `find_files_by_name` Lua-VM statt.

- Parameter: `suffixes` — Tabelle mit Zeichenketten (passende Pfadsuffixe)
- Gibt zurück: `{string, ...}` — übereinstimmende Dateipfade, ohne symbolische Links (kein Fehlertupel)

```
local pom_files = sbomgen.find_files_by_suffix({"/pom.properties"})
local release_headers = sbomgen.find_files_by_suffix({"ap_release.h", "openssl.h"})
```

sbomgen.find_files_by_path_regex (Muster)

Gibt Dateien zurück, deren durch Schrägstrich normalisierter Pfad mit einem der angegebenen Go-Regex-Muster (RE2) übereinstimmt. Der Abgleich erfolgt außerhalb der Lua-VM, was dies bei großen Dateilisten effizient macht.

- Parameter: `patterns` — Tabelle mit Go-Regex-Zeichenketten
- Gibt zurück: `{string, ...}` — übereinstimmende Dateipfade, ausgenommen symbolische Links (kein Fehlertupel)
- Löst aus: einen Lua-Fehler, wenn ein Muster nicht kompiliert werden kann

```
local configs = sbomgen.find_files_by_path_regex({"/etc/.*\\.conf$", "/opt/.*/config\
\.json$"})
```

Leistung: find_files_by_* vs get_file_list

Ziehen Sie bei Discovery-Plugins die Iteration in Lua vor. `find_files_by_name` `find_files_by_suffix` `find_files_by_path_regex` `get_file_list()` Bei einem Localhost-Scan mit 300.000 Dateien `string.match()` dauert die Iteration der Dateiliste in Lua mit ~15 Sekunden und ist in weniger als 1 Millisekunde abgeschlossen. `find_files_by_name` Der Unterschied besteht darin, dass jeder Dateipfad als Zeichenfolge in die Lua-VM `get_file_list()` kopiert wird. Anschließend interpretiert Lua die Schleife und den Mustervergleich für jeden einzelnen. Die `find_files_by_*` Funktionen führen den Abgleich außerhalb der Lua-VM durch und geben nur die übereinstimmenden Pfade zurück, wodurch sowohl der Kopier- als auch der Interpretationsaufwand pro Pfad vermieden wird.

Verwenden Sie diese `get_file_list()` Option nur, wenn Sie eine benutzerdefinierte Abgleichslogik benötigen, die nicht als Basisname, Suffix oder Regex-Übereinstimmung ausgedrückt werden kann.

sbomgen.read_dir (Pfad)

Listet Einträge in einem Verzeichnis auf.

- Gibt zurück: `{{name, is_dir}, ...}, err`

```
local entries, err = sbomgen.read_dir("/app/node_modules")
if err then return end
for _, e in ipairs(entries) do
    if e.is_dir then
        sbomgen.log_debug("directory: " .. e.name)
    end
end
```

sbomgen.file_stat (Pfad)

Gibt Metadaten zu einer Datei zurück.

- Gibt zurück: `{is_regular, is_dir, size}, err`

```
local info, err = sbomgen.file_stat(path)
if err then return end
if info.is_regular and info.size > 0 then
    -- process file
end
```

sbomgen.read_zip_entry (Pfad, Eintragspfad)

Liest einen einzelnen Eintrag aus einem ZIP-, JAR- oder WAR-Archiv.

- Gibt zurück: `string, err`

```
local manifest, err = sbomgen.read_zip_entry(
    "/app/lib/example.jar",
    "META-INF/MANIFEST.MF"
)
```

sbomgen.search_binary (Pfad, Regex)

Analysiert eine Datei als ELF, PE oder Mach-O Binärdatei und durchsucht den Standardabschnitt nach einer Go-Regex-Übereinstimmung. constant/variable

- Gibt zurück: `string|nil, err` — die übereinstimmende Zeichenfolge oder nil, wenn keine Übereinstimmung vorliegt

```
local version, err = sbomgen.search_binary(path, "Version:\\s+([\\d.]+)")
if version then
    sbomgen.log_info("found version: " .. version)
end
```

sbomgen.search_binary_all (Pfad, Regex [, n])

Analysiert eine Datei als ELF, PE oder Mach-O Binärdatei und gibt alle eindeutigen Treffer der ersten Capture-Gruppe aus dem Standardabschnitt zurück. constant/variable Übergebenn, um die Ergebnisse einzuschränken.

- Gibt zurück: `{string, ...}|nil, err` — Tabelle mit übereinstimmenden Zeichenketten oder Null, wenn keine Treffer gefunden wurden

```
local versions, err = sbomgen.search_binary_all(path, "version[= ]+([\\d.]+)", 5)
if versions then
    for _, v in ipairs(versions) do
        sbomgen.log_info("found: " .. v)
    end
end
```

sbomgen.search_binary_raw (Pfad, regulärer Ausdruck)

Durchsucht die gesamte Binärdatei nach der ersten Regex-Übereinstimmung und ist nicht auf einen bestimmten Abschnitt beschränkt. Wird verwendet, wenn die abschnittsbasierte Suche (`search_binary`) nicht ausreicht, z. B. wenn sich Versionszeichenfolgen in Abschnitten befinden, die nicht dem Standard entsprechen.

- Gibt zurück: `string|nil, err` — die übereinstimmende Zeichenfolge oder nil, wenn keine Übereinstimmung vorliegt

```
local version, err = sbomgen.search_binary_raw(path, "ProductVersion[\\x00\\s]+([\
\d.]+)")
```

FileHandle Methoden

FileHandle Objekte werden von zurückgegebensbomgen.`open_file()`.

`fh:read_line ()`

Liest die nächste Zeile (ohne das Zeilenumbruchzeichen). Kehrt `nil` bei EOF zurück.

- Gibt zurück: `string|nil, err`

`fh:read (n)`

Liest bis zu Byte. n Kehrt `nil` bei EOF zurück.

- Gibt zurück: `string|nil, err`

`fh:close ()`

Schließt das Datei-Handle. Schließt die Griffe immer, wenn Sie fertig sind.

Binäre Dienstprogramme

`sbomgen.hash (Daten, Algorithmus)`

Gibt den hexadezimalen Digest einer speicherinternen Bytezeichenfolge unter dem angegebenen Algorithmus zurück. Kombinieren Sie mit`sbomgen.read_file(path)`, um eine Datei zu hashen, die Sie bereits zum Parsen gelesen haben. Dies ist vorzuziehen, `sbomgen.hash_file(path)` wenn Sie sowohl die Bytes als auch den Digest benötigen, da die Datei nicht erneut gelesen wird. `hash`

- Gibt zurück: `string, err`
- Algorithmen: Die Liste der akzeptierten Algorithmuskonstanten finden Sie unterthe section called “Komponenten-Hashes”.

```
local data, err = sbomgen.read_file("/path/to/manifest.json")
if data then
    local sha256 = sbomgen.hash(data, sbomgen.hash_algorithms.SHA256)
    sbomgen.log_info("SHA-256: " .. sha256)
end
```

sbomgen.hash_file (Pfad, Algorithmus)

Gibt die hexadezimale Zusammenfassung des Inhalts einer Datei nach dem angegebenen Algorithmus zurück. Leitet den Lesevorgang durch die I/O Artefaktebene, sodass er für Verzeichnis-, Container-, Archiv-, Volume- und Localhost-Artefakte einheitlich funktioniert. Verwenden Sie diese Option, wenn der Digest das Einzige ist, was Sie aus der Datei benötigen.

- Gibt zurück: `string, err`

```
local sha256, err = sbomgen.hash_file("/app/bin/server",
 sbomgen.hash_algorithms.SHA256)
if sha256 then
    sbomgen.log_info("SHA-256: " .. sha256)
end
```

sbomgen.sha256 (Pfad)

> **Important**
>
> Als veraltet gekennzeichnet. Verwenden Sie stattdessen `sbomgen.hash_file(path, sbomgen.hash_algorithms.SHA256)`. Dieser Alias wird aus Gründen der Abwärtskompatibilität beibehalten und wird weiterhin funktionieren, er wird jedoch in einer future Version entfernt. Neue Plugins sollten aufgerufen werden, `hash_file` sodass die Wahl des Algorithmus explizit ist.

Äquivalent mit `sbomgen.hash_file(path, "SHA-256")`.

- Gibt zurück: `string, err`

```
local hash, err = sbomgen.sha256("/app/bin/server")
if hash then
    sbomgen.log_info("SHA-256: " .. hash)
end
```

sbomgen.contains_bytes (Pfad, Muster)

Prüft, ob eine Datei jedes der angegebenen Byte-Muster enthält. Gibt eine Tabelle mit booleschen Werten in derselben Reihenfolge wie die Eingabemuster zurück.

- Gibt zurück: `{bool, ...}, err`

```
local results, err = sbomgen.contains_bytes(path, {
    "\xff Go buildinf:",   -- Go build identifier
    "/rustc/",             -- Rust build identifier
})
if results then
    local is_go = results[1]
    local is_rust = results[2]
end
```

sbomgen.get_pe_version_info (Pfad)

Analysiert Windows PE-Versionsressourcen aus einer Binärdatei. Gibt eine Tabelle mit Versionsfeldern zurück, oder `nil, err` wenn die Datei keine PE-Binärdatei ist oder keine Versionsressource hat.

- Gibt zurück: `{product_version, file_version, string_table}, err`

Die `file_version` Felder `product_version` und stammen aus der `FixedFileInfo` PE-Struktur, formatiert als`"major.minor.build.revision"`. Das `string_table` Feld ist eine verschachtelte Tabelle, die mit einem Gebietsschema-Code versehen ist (z. B. `"040904B0"` für US-amerikanisches Unicode). Jedes Gebietsschema ist einer Tabelle mit name/value Paaren zugeordnet, die aus dem PE `StringFileInfo` (`ProductVersion`, `ProductNameFileDescription`, usw.) gezogen wurden. Eine PE-Binärdatei kann ein oder mehrere Gebietsschemas verfügbar machen.

```
local info, err = sbomgen.get_pe_version_info(file_path)
if err then return end

-- Fixed version fields (always flat)
local product_ver = info.product_version  -- e.g. "25.1.0.0"
local file_ver    = info.file_version     -- e.g. "25.1.0.0"
```

```
-- String table — iterate locales, or address a known locale by key
for locale, fields in pairs(info.string_table or {}) do
    sbomgen.log_info(string.format("%s ProductName=%s", locale, fields.ProductName or
 ""))
end

-- US English Unicode is the most common locale for PE files
local us = (info.string_table or {})["040904B0"]
if us then
    local display_ver = us.ProductVersion  -- e.g. "25.01"
    local name        = us.ProductName     -- e.g. "7-Zip"
end
```

sbomgen.parse_product_version (Pfad)

Praktischer Wrapper, der nur die Zeichenfolge der Produktversion aus einer PE-Binärdatei zurückgibt. FixedFileInfo Entspricht dem Aufrufen `get_pe_version_info(path)` und Lesen`product_version`.

- Gibt zurück: `string, err`

```
local version, err = sbomgen.parse_product_version(file_path)
if version then
    sbomgen.log_info("product version: " .. version)
end
```

sbomgen.parse_file_version (Pfad)

Praktischer Wrapper, der nur die Dateiversionszeichenfolge aus einer PE-Binärdatei zurückgibt. FixedFileInfo Entspricht dem Aufrufen `get_pe_version_info(path)` und Lesen`file_version`.

- Gibt zurück: `string, err`

```
local version, err = sbomgen.parse_file_version(file_path)
if version then
    sbomgen.log_info("file version: " .. version)
end
```

Paket-Ausgabe

`sbomgen.push_package (Paket)`

Verschiebt eine Paketsuche in die SBOM. Nur in Sammlungs-Plugins verfügbar.

Die `pkg` Tabelle unterstützt die folgenden Felder:

Feld	Typ	Erforderlich	Beschreibung
`name`	Zeichenfolge	Ja	Paketname, der als Komponentenname in der SBOM verwendet wird
`purl_name`	Zeichenfolge	Nein	PURL-Name nskomponente; standardmäßig. `name` Legt diesen Wert fest, wenn der Anzeigena me kein guter Paketbezeichner ist, z. B. mit `name = "Visual Studio Code" purl_name = "visualst udiocode"`
`version`	Zeichenfolge	Nein	Aufgelöste Versionsz eichenfolge
`namespace`	Zeichenfolge	Nein	PURL-Namespace (z. B.,,`"curl"`) `"wordpress/ plugin"`
`purl_type`	Zeichenfolge	Ja	PURL-Typ (z. B.,`"pypi"`,, `"npm"` `"cargo""deb","generic"`

Feld	Typ	Erforderlich	Beschreibung
			) oder `sbomgen.purl_types.NONE` für Komponenten, die keine Paket-URL enthalten (siehe) the section called “Konstanten vom Typ PURL”
`component_type`	Zeichenfolge	Ja	CyclonedX-Komponentyp; verwendet `sbomgen.component_types.*` Konstanten (z. B.) `sbomgen.component_types.LIBRARY`
`qualifiers`	Tabelle	Nein	PURL-Qualifizierer als Schlüssel-Wert-Paare (erscheinen in der Paket-URL)
`properties`	Tabelle	Nein	Eigenschaften von CyclonedX-Komponenten als Schlüssel-Wert-Paare (siehe) the section called “Eigenschaften von CyclonedX”

Feld	Typ	Erforderlich	Beschreibung
`hashes`	Tabelle	Nein	Komponenten-Hashes, die nach dem Namen des Algorithmus eingegeben sind; siehe the section called “Komponenten-Hashes”
`licenses`	Tabelle	Nein	Liste von Lizenztexten, normalisiert auf einen SPDX-Ausdruck; siehe the section called “Komponentenlizenzen”
`children`	Tabelle	Nein	Verschachtelte untergeordnete Pakete, jedes mit derselben Form wie `pkg` (Pflichtfelder werden rekursiv überprüft)

Feld	Typ	Erforderlich	Beschreibung
`source_paths`	Tabelle	Nein	Liste der als Eigenschaften aufgezeichneten Herkunftspfade. `source_path` Standardmäßig ist die Datei voreingestellt, mit der aufgerufen `collect()` wurde. Stellen Sie es so ein, dass es auf ein Verzeichnis, auf mehrere Pfade zeigt oder einem Kind einen eigenen Pfad gibt
`provenance`	Zeichenfolge	Nein	Woher das Paket gelesen wurde; verwenden Sie `sbomgen.provenance.*` Konstanten. Standardeinstellung: `sbomgen.provenance.FILE` . Siehe the section called “Herkunft des Package”

```
sbomgen.push_package({
    name = "requests",
    version = "2.28.1",
    purl_type = "pypi",
```

```
    component_type = sbomgen.component_types.LIBRARY,
    qualifiers = { example_qualifier = "example_qualifier_value" },
    properties = {
        -- Use your own namespace; amazon:inspector:* is reserved for Amazon Inspector.
        ["acme:example:extra_field"] = "example_value",
    },
    hashes = {
        [sbomgen.hash_algorithms.SHA256] =
 "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824",
    },
})
```

Herkunft des Package

Die meisten Pakete werden durch Lesen einer Datei gefunden, und die SBOM zeichnet diese Datei als Herkunft der Komponente auf. Bei einigen ist das nicht der Fall: Eine Windows Anwendung kann nur durch einen Registrierungsschlüssel beschrieben werden, ohne dass auf der Festplatte ihre Version benannt ist.

Das `provenance` Feld gibt an, welcher Fall zutrifft.

Wert	Herkunft aufgezeichnet
`sbomgen.provenance.FILE` (Standard)	`source_file_scanner` plus ein `source_path` pro Eintrag in`source_paths`, wobei standardmäßig die empfangene Datei verwendet wird `collect()`
`sbomgen.provenance.REGISTRY`	`source_registry_scanner`, und nein`source_path`, weil keine Datei das Paket beschreibt

Ein unbekannter Benutzer `provenance` löscht das Paket mit einer Warnung, anstatt im Hintergrund auf die Herkunft der Datei zurückzugreifen.

```
sbomgen.push_package({
    name = "Example App",
    purl_name = "exampleapp",
    version = "1.2.3",
    namespace = "vendor",
```

```
    purl_type = "generic",
    component_type = sbomgen.component_types.APPLICATION,
    provenance = sbomgen.provenance.REGISTRY,
    properties = {
        [sbomgen.properties.WINDOWS_REGISTRY_KEY] = registry_key,
    },
})
```

Sammeln ohne gefundene Datei

Ein Collector wird normalerweise einmal pro erkannter Datei ausgeführt, sodass ein Ereignis ohne Dateien ihn nie erreicht. Ein Discovery-Plugin kann Ereignisse auflisten, die ohnehin ausgelöst werden sollten, und zwar für Collectoren, die die Registrierung oder eine andere Quelle lesen, die keine Dateiquelle ist:

```
function get_fileless_events()
    return { "FoundExampleApp" }
end
```

`collect()`wird dann genau einmal für dieses Ereignis mit einem leeren `file_path` Wert ausgeführt. Ereignisse, die nicht aufgeführt sind, bleiben stumm, wenn die Erkennung nichts findet, sodass die Rückgabe einer leeren Tabelle von ihre übliche Bedeutung `discover()` beibehält.

Gruppieren von Ergebnissen nach Dateien

`collect()`sieht jeweils eine Datei, sodass ein Plugin, das ein Paket pro Aufruf weiterleitet, eine separate Komponente pro Datei ausgibt. Wenn mehrere Dateien eine Installation derselben Sache beschreiben, ist das normalerweise falsch: Vier Kopien einer in derselben Version ausführbaren Browserdatei sind eine Komponente mit vier Quellpfaden, nicht vier Komponenten.

Accumulate across calls and emit from`finalize()`, das einmal nach der letzten Datei des Events ausgeführt wird:

```
local pending = {}

function collect(file_path)
    local version = sbomgen.parse_product_version(file_path)
    if not version then
        return
    end
    pending[version] = pending[version] or {}
```

```
    table.insert(pending[version], dir_name(file_path))
end

function finalize()
    local batch = pending
    pending = {}
    for version, paths in pairs(batch) do
        sbomgen.push_package({
            name = "Example App",
            version = version,
            purl_type = "generic",
            component_type = sbomgen.component_types.APPLICATION,
            source_paths = paths,
        })
    end
end
```

Löscht den Akkumulator vor dem Senden, wie oben beschrieben, nicht danach. Der Lua Status eines Plug-ins bleibt während des gesamten Scans bestehen, sodass alles, was zurückbleibt, in das nächste Ereignis übergeht, und wenn es zuerst gelöscht wird, bedeutet, dass ein Fehler während der Ausgabe keine Einträge mehr enthalten kann.

Komponenten-Hashes

Das optionale `hashes` Feld für Digests zur Integrität von `sbomgen.push_package()` Datensätzen für eine Komponente. Die Einträge werden nach dem Namen des Algorithmus eingegeben und in das `components[].hashes` CyclonedX-Array serialisiert, sodass sie dem Schema entsprechen, das Amazon Inspector erwartet.

Unterstützte Algorithmen

Verwenden Sie die Konstanten unter, `sbomgen.hash_algorithms` sodass der ans`bomgen.hash()`/übergebene Algorithmusname dieselbe Zeichenfolge ist, die akzeptiert `sbomgen.hash_file()` wird von: `push_package({ hashes = ... })`

Konstante	Wert	Länge des Hex-Digests
`sbomgen.hash_algorithms.SHA1`	`"SHA-1"`	40

Konstante	Wert	Länge des Hex-Digests
`sbomgen.hash_algorithms.SHA256`	`"SHA-256"`	64

Regeln für die Validierung

`push_package()`validiert, `hashes` bevor ein Ergebnis ausgegeben wird. Ein Paket, dessen Hashes bei der Validierung fehlschlagen, wird gelöscht und eine Warnung wird protokolliert. Validierung:

- Algorithmusnamen müssen mit einem Eintrag in übereinstimmen `sbomgen.hash_algorithms` (Groß- und Kleinschreibung beachten, genau `"SHA-1"` oder`"SHA-256"`).
- Bei den Werten darf es sich um nicht leere Hexadezimalziffern in Kleinbuchstaben handeln.
- Die Werte müssen die richtige Länge für den Algorithmus haben (40 Zeichen für, 64 Zeichen für SHA-1). SHA-256
- Die Validierung erfolgt rekursiv: Ein fehlerhafter Hash im Inneren `children[].hashes` lehnt das gesamte Paket ab.

Beispiel: Ein Manifest als Hashwert speichern und den Digest anhängen

```
function collect(file_path)
    local data, err = sbomgen.read_file(file_path)
    if err then return end

    local sha256 = sbomgen.hash(data, sbomgen.hash_algorithms.SHA256)

    sbomgen.push_package({
        name = "skill-manifest",
        version = "1.0.0",
        purl_type = "generic",
        component_type = sbomgen.component_types.DATA,
        hashes = {
            [sbomgen.hash_algorithms.SHA256] = sha256,
        },
    })
end
```

Wenn der Digest das Einzige ist, was Sie von der Datei benötigen, `sbomgen.hash_file(path, algo)` ziehen Sie es vor, die Datei zweimal zu lesen — es leitet den Lesevorgang in einem einzigen Durchgang durch die I/O Artefaktebene.

Komponentenlizenzen

Das optionale `licenses` Feld für `sbomgen.push_package()` zeichnet die Lizenzen auf, die ein Paket deklariert. Übergibt die Lizenztexte so, wie Sie sie lesen. sbomgen normalisiert sie zu einem einzigen SPDX-Lizenzausdruck und serialisiert ihn in das CyclonedX-Array. `components[].licenses`

```
sbomgen.push_package({
    name = "docker",
    version = "29.7.1",
    purl_type = "brew",
    component_type = sbomgen.component_types.APPLICATION,
    licenses = {"Apache-2.0"},
})
```

Produziert:

```
{
  "type": "application",
  "name": "docker",
  "version": "29.7.1",
  "purl": "pkg:brew/docker@29.7.1",
  "licenses": [ { "expression": "Apache-2.0" } ]
}
```

Normalisierung

Werte müssen keine gültigen SPDX-Bezeichner sein. sbomgen normalisiert sie, sodass ein Plugin alles weitergeben kann, was in einem Paketmanifest deklariert ist:

Eingabe	Resultierender Ausdruck
`{"Apache-2.0"}`	`Apache-2.0`
`{"GPL3+"}`	`GPL-3.0-or-later`

Eingabe	Resultierender Ausdruck
`{"MIT", "Apache-2.0"}`	`Apache-2.0 AND MIT`
`{"Apache-2.0", "Apache-2.0"}`	`Apache-2.0` (dedupliziert)
`{"Custom Vendor License"}`	`LicenseRef-Custom-Vendor-License`

Mehrere Einträge werden kombiniert mit. `AND` Text, der kein anerkannter SPDX-Bezeichner ist, wird zu einem `LicenseRef-` Ausdruck, anstatt gelöscht zu werden. Leere Zeichenketten werden ignoriert.

Lizenzen werden nur ausgestellt, wenn der Benutzer danach fragt

Lizenzen erscheinen nur dann in der SBOM, wenn der Scan mit ausgeführt wurde. `--collect-licenses` Ohne diese Markierung wird das Feld weggelassen, genau wie bei den eingebauten Kollektoren von Sbomgen.

Das Lesen von Lizenzdaten kann teuer sein, zum Beispiel wenn dafür eine separate Lizenzdatei pro Paket geöffnet werden muss. Überprüfen Sie diese Arbeit `sbomgen.should_collect_licenses()` und überspringen Sie sie, wenn sie nicht benötigt wird:

```
function collect(file_path)
    local licenses = nil
    if sbomgen.should_collect_licenses() then
        licenses = read_license_file(file_path)
    end

    sbomgen.push_package({
        name = "example",
        version = "1.0.0",
        purl_type = "generic",
        component_type = sbomgen.component_types.APPLICATION,
        licenses = licenses,
    })
end
```

Es ist auch in Ordnung, `licenses` bedingungslos zu bestehen. Das Feld wird einfach ignoriert, wenn die Lizenzerfassung deaktiviert ist.

Pakete für Kinder

Ein Paket `children` trägt sein eigenes`licenses`. Lizenzen werden nicht vom übergeordneten Paket übernommen, da ein verschachteltes Paket möglicherweise anders lizenziert wird als die Komponente, unter der es ausgeliefert wird.

Eigenschaften von CyclonedX

CyclonedX-Eigenschaften sind Schlüsselwertmetadaten, die an eine Komponente in der SBOM angehängt sind. Sie unterscheiden sich von PURL-Qualifizierern:

- **`qualifiers`**— PURL-Qualifizierer. Diese werden Teil der Paket-URL-Zeichenfolge (z. B.`pkg:deb/debian/curl@7.88.1?arch=amd64`). Einige PURL-Qualifier haben für Amazon Inspector eine semantische Bedeutung und beeinflussen die Identifizierung von Sicherheitslücken. Weitere Informationen finden Sie unter Was ist eine Paket-URL? für die spezifischen Konventionen von Inspector.
- **`properties`**— Eigenschaften der CyclonedX-Komponente. Diese erscheinen im `components[].properties` Array der SBOM und ändern nicht, wie die Komponente identifiziert wird.

Reservierte Namespaces

Die `amazon:inspector:*` Familie der CyclonedX-Eigenschaftsnamespaces ist Amazon Inspector vorbehalten:

- `amazon:inspector:sbom_generator:*`— wird von sbomgen und seinen eingebauten Scannern verwendet.
- `amazon:inspector:sbom_scanner:*`— wird von der Amazon Inspector Scan API verwendet.

Plugin-defined Eigenschaften dürfen diese Namespaces nicht verwenden. Das Schreiben in einen reservierten Namespace kann Werte, auf die sich Inspector stützt, überlagern oder mit ihnen in Konflikt geraten, und die resultierende SBOM kann bei der Identifizierung von Sicherheitslücken falsch interpretiert werden. Eine vollständige Liste der reservierten Schlüssel finden Sie unter Verwenden von CyclonedX-Namespaces mit Amazon Inspector.

Regeln zur Benennung von Schlüsseln

Eigenschaftsschlüssel, die an übergeben wurden, `sbomgen.push_package()` werden wie folgt verarbeitet:

Eingabeschlüssel	Resultierender Schlüssel in SBOM	Für benutzerdefinierte Plugins empfohlen?
Enthält `:` (z. B.`acme:my_plugin:field`)	Wird wörtlich verwendet	Ja — platziere jede vom Plugin definierte Eigenschaft in deinem eigenen Namespace
Nein `:` (z. B.,) `field`	Auto-prefixed zu `amazon:inspector:sbom_generator:field`	Nein — das schreibt in einen reservierten Namespace

Fügen Sie in Eigenschaftsschlüsseln, die Sie definieren, immer mindestens einen Doppelpunkt ein. Verwenden Sie einen Namespace, der für Ihre Organisation oder Ihr Plugin einzigartig ist (zum Beispiel`acme:python-pip:*`):

```
properties = {
    -- Custom namespace — safe to use (recommended)
    ["acme:python-pip:manifest_path"] = file_path,
    ["acme:python-pip:pinned"]        = "true",

    -- Fully-qualified key outside amazon:inspector:* — also fine
    ["my:custom:namespace:key"] = "value",

    -- No colon: avoid — ends up as "amazon:inspector:sbom_generator:custom_field"
    -- custom_field = "value",
}
```

Von sbomgen festgelegte Eigenschaften

Sbomgen kann jeder Komponente, die es ausgibt, eigene Eigenschaften zuweisen. Diese Werte stammen aus dem reservierten `amazon:inspector:sbom_generator:*` Namespace und sollten nicht von Plugins erzeugt werden. Beobachtetes Laufzeitverhalten:

- `source_path`wird immer von sbomgen hinzugefügt.
- `source_file_scanner`und `source_package_collector` werden hinzugefügt, wenn es aktiviert `--enable-debug-props` ist.

Die vollständige Taxonomie der reservierten Schlüssel finden Sie im Amazon Inspector-Benutzerhandbuch: Verwenden von CyclonedX-Namespaces mit Amazon Inspector.

Der SBOM-Inhalt wird nicht bereinigt

Sbomgenuntersucht oder filtert die Daten, die ein Plugin ausgibt, nicht. Komponentennamen, Versionen, PURLs, Hashes und Eigenschaftswerte werden wie angegeben in die SBOM geschrieben. Sbomgenerkennt oder redigiert keine geheimen Daten, Anmeldeinformationen, Token oder andere sensible Daten. Wenn ein Plugin einen solchen Wert in ein Ergebnis einfügt, erscheint er in der Ausgabe-SBOM und wird überall dort übertragen, wo diese SBOM veröffentlicht wird.

Sie sind dafür verantwortlich, was Ihre Plugins schreiben. Geben Sie nur Daten aus, die von dem Artefakt stammen, das Sie inventarisieren möchten, und behandeln Sie die SBOM bei der Entscheidung, was Sie aufnehmen möchten, als gemeinsam nutzbares Artefakt.

Eigenschaftskonstanten

Built-in Eigenschaftsschlüsselkonstanten sind über verfügbar. `sbomgen.properties`
Jede der folgenden Konstanten wird zu einem Schlüssel innerhalb des reservierten `amazon:inspector:sbom_generator:*` Namespaces aufgelöst. Diese Konstanten existieren, sodass die integrierten Scanner von sbomgen konsistente Eigenschaftsschlüssel ausgeben. Sie sind keine Erweiterungspunkte für benutzerdefinierte Plugins — wenn sie in einem benutzerdefinierten Plugin verwendet werden, wird in einen reservierten Namespace geschrieben, der Werte überschatten kann, auf die sich Inspector stützt. Siehe the section called “Reservierte Namespaces” oben.

Autoren benutzerdefinierter Plugins sollten Eigenschaften (zum Beispiel`acme:my_plugin:*`) in ihrem eigenen Namespace definieren, anstatt diese Konstanten wiederzuverwenden.

Konstante	Gelöster Wert
`sbomgen.properties.NAMESPACE`	`amazon:inspector:sbom_generator:`

Konstante	Gelöster Wert
sbomgen.properties.VENDOR	amazon:inspector:sbom_generator:vendor
sbomgen.properties.FILE_SIZE_BYTES	amazon:inspector:sbom_generator:file_size_bytes
sbomgen.properties.KERNEL_COMPONENT	amazon:inspector:sbom_generator:kernel_component
sbomgen.properties.RUNNING_KERNEL	amazon:inspector:sbom_generator:running_kernel
sbomgen.properties.UNRESOLVED_VERSION	amazon:inspector:sbom_generator:unresolved_version
sbomgen.properties.TRANSITIVE_DEPENDENCY	amazon:inspector:sbom_generator:experimental:transitive_dependency
sbomgen.properties.GO_REPLACE_DIRECTIVE	amazon:inspector:sbom_generator:replaced_by
sbomgen.properties.DUPLICATE_PACKAGE	amazon:inspector:sbom_generator:is_duplicate_package
sbomgen.properties.DUPLICATE_PURL	amazon:inspector:sbom_generator:duplicate_purl
sbomgen.properties.DOCKERFILE_CHECK	amazon:inspector:sbom_generator:dockerfile_finding
sbomgen.properties.CERTIFICATE_FINDING	amazon:inspector:sbom_generator:certificate_finding
sbomgen.properties.CERTIFICATE_SUBJECT_NAME	amazon:inspector:sbom_generator:certificate:subject_name

Konstante	Gelöster Wert
sbomgen.properties.CERTIFICATE_ISSUER_NAME	amazon:inspector:sbom_generator:certificate:issuer_name
sbomgen.properties.CERTIFICATE_SIGNATURE_ALGORITHM	amazon:inspector:sbom_generator:certificate:signature_algorithm
sbomgen.properties.CERTIFICATE_NOT_VALID_BEFORE	amazon:inspector:sbom_generator:certificate:not_valid_before
sbomgen.properties.CERTIFICATE_NOT_VALID_AFTER	amazon:inspector:sbom_generator:certificate:not_valid_after
sbomgen.properties.WINDOWS_REGISTRY_KEY	amazon:inspector:sbom_generator:registry_key
sbomgen.properties.SUBSCRIPTION_ENABLED	amazon:inspector:sbom_generator:subscription:enabled
sbomgen.properties.SUBSCRIPTION_NAME	amazon:inspector:sbom_generator:subscription:name
sbomgen.properties.SUBSCRIPTION_LOCKED_VERSION	amazon:inspector:sbom_generator:subscription:locked_version
sbomgen.properties.OPENSSL_FULL_VERSION	amazon:inspector:sbom_generator:openssl:full_version
sbomgen.properties.HARDENED_IMAGE_VENDOR	amazon:inspector:sbom_generator:hardened_image:vendor

Scanner-Gruppen

Discovery-Plugins müssen ihre Scanner-Gruppen über deklarierenget_scanner_groups(). Gruppen kategorisieren Scanner und ermöglichen es Benutzern, Kategorien selektiv zu aktivieren oder zu deaktivieren. Konstanten sind verfügbar über: sbomgen.groups

Konstante	Wert	Beschreibung
sbomgen.groups.OS	"os"	Betriebssystem-Paketmanager (dpkg, rpm usw.)
sbomgen.groups.PROGRAMMING_LANGUAGE	"programming-language-packages"	Sprachpaketmanager (pip, npm, maven usw.)
sbomgen.groups.BINARY	"binary"	Kompilierte Binäranalyse (Go, Rust)
sbomgen.groups.PACKAGE_COLLECTOR	"pkg-scanner"	Allgemeine Paketabholung
sbomgen.groups.EXTRA_ECOSYSTEMS	"extra-ecosystems"	Zusätzliche Ökosysteme (Curl, Nginx usw.)
sbomgen.groups.CERTIFICATE	"certificate"	Scannen von Zertifikaten
sbomgen.groups.CUSTOM	"custom"	Wird automatisch zu allen benutzerdefinierten Plugins hinzugefügt, die über geladen wurden --plugin-dir
sbomgen.groups.MACHINE_LEARNING	"machine-learning"	Modellerkennung durch maschinelles Lernen

Beispiel:

```
function get_scanner_groups()
    return {sbomgen.groups.PROGRAMMING_LANGUAGE, sbomgen.groups.PACKAGE_COLLECTOR}
```

```
end
```

Konstanten für den Komponententyp

Das `component_type` Feld in `push_package()` muss einer der CyclonedX 1.5-Komponententypen sein. Konstanten sind verfügbar über: `sbomgen.component_types`

Konstante	Wert
`sbomgen.component_types.APPLICATION`	`"application"`
`sbomgen.component_types.FRAMEWORK`	`"framework"`
`sbomgen.component_types.LIBRARY`	`"library"`
`sbomgen.component_types.CONTAINER`	`"container"`
`sbomgen.component_types.PLATFORM`	`"platform"`
`sbomgen.component_types.OPERATING_SYSTEM`	`"operating-system"`
`sbomgen.component_types.DEVICE`	`"device"`
`sbomgen.component_types.DEVICE_DRIVER`	`"device-driver"`
`sbomgen.component_types.FIRMWARE`	`"firmware"`
`sbomgen.component_types.FILE`	`"file"`
`sbomgen.component_types.MACHINE_LEARNING_MODEL`	`"machine-learning-model"`
`sbomgen.component_types.DATA`	`"data"`

Beispiel:

```
sbomgen.push_package({
    name = "requests",
    version = "2.28.1",
    purl_type = "pypi",
    component_type = sbomgen.component_types.LIBRARY,
})
```

Konstanten vom Typ PURL

`purl_type`ist normalerweise eine formlose Zeichenfolge aus der PURL-Spezifikation. Der eine Aufzählungswert ist verfügbar über: `sbomgen.purl_types`

Konstante	Wert	Beschreibung
`sbomgen.purl_types .NONE`	`"none"`	Gibt die Komponente ohne Paket-URL aus

Verwenden Sie diese Option`NONE`, wenn kein PURL-Typ die Komponente aussagekräftig identifiziert, z. B. wenn auf der Festplatte eine Modelldatei für maschinelles Lernen gefunden wurde. Die resultierende SBOM-Komponente hat kein Feld. `purl`

`purl_type`bleibt erforderlich. Wenn Sie es weglassen oder den Feldnamen falsch schreiben, wird das Paket mit einer Warnung gelöscht, anstatt im Hintergrund eine Komponente ohne Paket-URL zu erzeugen.

```
sbomgen.push_package({
    name = "model.gguf",
    purl_type = sbomgen.purl_types.NONE,
    component_type = sbomgen.component_types.MACHINE_LEARNING_MODEL,
})
```

Plattformkonstanten

Konstanten zum Vergleich mit. `sbomgen.get_platform()` Verfügbar über`sbomgen.platform`:

Konstante	Wert
`sbomgen.platform.LINUX`	`"linux"`
`sbomgen.platform.WINDOWS`	`"windows"`
`sbomgen.platform.DARWIN`	`"darwin"`

Beispiel:

```
if sbomgen.get_platform() == sbomgen.platform.WINDOWS then
    -- Windows-specific logic
end
```

Konstanten für den Hash-Algorithmus

Konstanten für den Algorithmusparameter von `sbomgen.hash()``sbomgen.hash_file()`, und das `hashes` Feld von. `sbomgen.push_package()` Die Zeichenkettenwerte stimmen mit den Namen des CyclonedX-Hash-Algorithmus überein, sodass dieselbe Konstante den gesamten Hash-Pfad ohne Übersetzung durchläuft.

Konstante	Wert
`sbomgen.hash_algorithms.SHA1`	`"SHA-1"`
`sbomgen.hash_algorithms.SHA256`	`"SHA-256"`

Beispiel:

```
local digest = sbomgen.hash_file(path, sbomgen.hash_algorithms.SHA256)
sbomgen.push_package({
    name = "example",
    purl_type = "generic",
    component_type = sbomgen.component_types.LIBRARY,
    hashes = { [sbomgen.hash_algorithms.SHA256] = digest },
})
```

Informationen zum Artifact

`sbomgen.get_platform ()`

Gibt die Zeichenfolge der Laufzeitplattform zurück (z. B.,,,). `"linux"` `"windows"` `"darwin"`

`sbomgen.get_artifact_type ()`

Gibt den Typ des Artefakts zurück, das gescannt wird (z. B.,). `"directory"` `"archive"`

`sbomgen.should_collect_licenses ()`

Gibt zurück`true`, ob der Benutzer die Lizenzerfassung über aktiviert hat. `--collect-licenses`

`sbomgen.get_env_vars ()`

Gibt Umgebungsvariablen aus dem Artefakt als Tabelle mit Einträgen zurück. `{key, value}`

```
local env_vars = sbomgen.get_env_vars()
for _, env in ipairs(env_vars) do
    if env.key == "NODE_ENV" then
        sbomgen.log_info("Node environment: " .. env.value)
    end
end
```

`sbomgen.get_system_drive ()`

Gibt den Systemlaufwerksbuchstaben (z. B.) aus der Umgebung des Artefakts zurück. `"C:"` Liest die `SystemDrive` Umgebungsvariable und verwendet standardmäßig, `"C:"` falls sie nicht gesetzt ist. Dies ist das Lua-Äquivalent von. `strutils.GetSystemDriverLetter()`

```
local drive = sbomgen.get_system_drive()
local program_files = drive .. "/Program Files/"
```

`sbomgen.resolve_glob_paths (Muster)`

Erweitert die Glob-Muster des Dateisystems anhand des Host-Dateisystems. Localhost-only: gibt bei anderen `nil` Artefakttypen einen Fehler zurück.

```
function get_localhost_scan_paths()
    return sbomgen.resolve_glob_paths({
        "/home/*/.cache/huggingface/hub",
```

```
        "/Users/*/.cache/huggingface/hub",
        "C:/Users/*/.cache/huggingface/hub",
    })
end
```

Verhalten:

- Die Mustersyntax folgt der von Go `filepath.Match`:`*`,`?`,`[abc]`,`[a-z]`.
- Eingabemuster und Ausgabepfade sind normalisiert: Redundante Trennzeichen (`a//b`), Punktsegmente (`a/./b`) und nachfolgende Trennzeichen (`a/b/`) werden ausgeblendet.
- Die Ausgabe ist dedupliziert; das erste Vorkommen eines Pfads gewinnt. Die Reihenfolge der Eingabemuster wird im gesamten Ergebnis beibehalten.
- Muster, die mit nichts übereinstimmen, geben keine Einträge zurück. Empty-string Muster werden stillschweigend übersprungen. Fehlerhafte Muster (z. B. nicht übereinstimmende Klammern) geben eine Warnung aus und werden übersprungen.

Cross-platform Pfadtrennzeichen:

- Verwenden Sie Schrägstriche (`/`) für alle Pfade. Schrägstriche funktionieren unter Linux, macOS und Windows; die Dateipfadlogik von Go übersetzt sie in das native Trennzeichen unter Windows.
- Backslash-Trennzeichen funktionieren nur unter Windows. Unter Linux und macOS `\` ist es ein wörtliches Dateinamenzeichen, kein Pfadtrennzeichen — ein Muster, wie es auf POSIX-Systemen nichts `"C:\\Users\\*"` passt.
- Vermeiden Sie wörtliche Windows-style Pfade in Lua-Strings. Eine Lua-Zeichenfolge wie `"C:\Users"` wird so interpretiert, als ob `C:<form-feed>sers` sie kein gültiges Lua-Escape `\U` ist (und `\t` usw. sind) `\f\n`, sodass das Muster stillschweigend fehlschlägt. Verwenden Sie entweder Schrägstriche, maskierte Backslashes (`"C:\\Users"`) oder eine unformatierte Zeichenfolge mit langen Klammern (). `[[C:\Users]]`

`sbomgen.get_home_dirs ()`

Gibt die Stammverzeichniswurzeln des Benutzers für das Artefakt als Pfadtabelle zurück. Localhost zählt das echte Host-Dateisystem auf; Container- und Volume-Artefakte zählen ihr eigenes (rootfs- oder gemountetes) Dateisystem auf; alle anderen Artefakttypen geben eine leere Tabelle zurück. Pfade werden mit Schrägstrichen normalisiert, dedupliziert und sortiert.

Dies ist die Methode, die Artefakte berücksichtigt, um Benutzerverzeichnisse (z. B. Modell-Caches oder Werkzeugkonfigurationen) zu finden, ohne fest codieren zu müssen oder. `/home/*` `/Users/*` Es beinhaltet das Home-Verzeichnis von Root und überspringt bekannte Nicht-Benutzerverzeichnisse (integrierte Profile wie `Public` and on, on und `Default` on). Windows `Shared` macOS `lost +found` Linux

```
function get_localhost_scan_paths()
    local patterns = {}
    for _, home in ipairs(sbomgen.get_home_dirs()) do
        table.insert(patterns, home .. "/.cache/huggingface/hub")
    end
    return sbomgen.resolve_glob_paths(patterns)
end
```

Die obige Zusammensetzung bezieht sich nur auf Localhost, da `resolve_glob_paths` Fehler bei Artefakten auftreten, die nicht auf Localhost basieren. Ordnen Sie bei Container- und Volume-Scans stattdessen die zurückgegebenen Home-Roots mit der Dateiliste des Artefakts ab (z. B. mit). `sbomgen.find_files_by_path_regex`

Verhalten:

- Definiert für `localhostcontainer`, und `volume` Artefakte. Andere Artefakttypen geben eine leere Tabelle zurück, sowohl weil ein benutzerspezifisches Home-Konzept nicht für ein bloßes Verzeichnis oder einen Binärscan gilt, als auch weil das Lesen absoluter Hostpfade dort das Scan-Root-Verzeichnis umgehen würde.
- Gibt auf localhost absolute Hostpfade zurück (z. B.,`/home/alice`). `/root` Gibt Pfade auf einem Container oder Volume so zurück, wie sie im Dateisystem dieses Artefakts vorkommen.
- Bei der Aufzählung wird die Artefaktschnittstelle gelesen, sodass Container- und Volume-Homes anhand ihres eigenen Dateisystems und nicht anhand des Hosts aufgelöst werden.
- Einträgen mit Symlinks wird nicht gefolgt; es werden nur echte Verzeichnisse zurückgegeben.
- OnWindows, das `Users` Verzeichnis befindet sich auf dem Verzeichnis des Scan-Hosts`SystemDrive`, sodass ein Windows Container oder ein Volume unter dem Laufwerksbuchstaben des Hosts aufgeführt wird. Dies ist eine bekannte Einschränkung, die gemeinsam mit verwendet wird. `sbomgen.get_system_drive`

Informationen zum System

Diese Funktionen geben Metadaten über das Betriebssystem und die Hardware des Artefakts zurück. Werte können leere Zeichenketten sein, wenn die Informationen nicht verfügbar sind (z. B. beim Scannen eines Verzeichnisses ohne Betriebssystem-Metadaten).

Funktion	Rückgabewerte
`sbomgen.get_os_name()`	Name des Betriebssystems (z. B.,`"Ubuntu"`,`"Alpine Linux"`)
`sbomgen.get_os_version()`	Betriebssystemversion (z. B.,`"22.04"`,`"3.18"`)
`sbomgen.get_os_codename()`	Betriebssystem-Codename (z. B.,,`"jammy"`) `"bookworm"`
`sbomgen.get_os_id()`	Betriebssystem-ID (z. B.,,`"ubuntu"`) `"alpine"`
`sbomgen.get_kernel_name()`	Kernelname (z. B.`"Linux"`)
`sbomgen.get_kernel_version()`	Zeichenfolge für die Kernel-Version
`sbomgen.get_cpu_arch()`	CPU-Architektur (z. B.,`"x86_64"`,`"aarch64"`)
`sbomgen.get_hostname()`	Hostname des Systems

Reguläre Ausdrücke

Den integrierten Mustern von Lua fehlen Funktionen wie Alternation (`|`), Quantifier Ranges (`{n,}`) und Lookahead. Um diese Lücke zu schließen, stellt sbomgen das Paket von Go direkt zur Verfügung. `regexp` Diese Funktionen verwenden die Go-Regex-Syntax (RE2), keine Lua-Muster.

`sbomgen.regex_find (str, Muster)`

Gibt die erste Übereinstimmung mit einem Go-Regex-Muster zurück, oder wenn es keine Übereinstimmung gibt. `nil`

- Gibt zurück: `string|nil, err`

```
local version = sbomgen.regex_find(content, "\\d+\\.\\d+\\.\\d+")
```

sbomgen.regex_match (str, Muster)

Gibt Capture-Gruppen aus dem ersten Treffer zurück. Index 1 ist die vollständige Übereinstimmung, 2+ sind Capture-Gruppen.

- Gibt zurück: `{string, ...}|nil, err`

```
local groups = sbomgen.regex_match(content, "(MySQL|MariaDB) (\\d+)\\.(\\d+)\\.(\\d+)")
if groups then
    local db_type = groups[2]   -- "MySQL" or "MariaDB"
    local major   = groups[3]
end
```

sbomgen.regex_find_all (str, Muster [, n])

Gibt alle Treffer zurück, die sich nicht überschneiden. An n das Limit der Ergebnisse übergeben (Standardeinstellung: alle).

- Gibt zurück: `{string, ...}|nil, err`

```
local versions = sbomgen.regex_find_all(content, "\\d+\\.\\d+\\.\\d+")
```

sbomgen.regex_replace (str, Muster, Ersatz)

Ersetzt alle Treffer. Die Ersatzzeichenfolge kann $1$2, usw. für Capture-Gruppenverweise verwenden.

- Gibt zurück: `string, err`

```
local cleaned = sbomgen.regex_replace(raw_version, "(1[6-9]\\d{8,}|
buildkitsandbox.*)$", "")
```

Wann sollten Regex- und Lua-Muster verwendet werden

Verwenden Sie das in Lua integrierte`string.match`/`string.find`für einfache Muster — sie sind schneller und erfordern keine maskierten Backslashes. Verwenden Sie `sbomgen.regex_*` es, wenn Sie Folgendes benötigen:

- Abwechslung: `(foo|bar)`
- Bereiche des Quantifizierers: `\d{8,}`
- Komplexe Zeichenklassen, die in Lua-Mustern nicht ausgedrückt werden können

Strukturiertes Parsen

Sbomgen stellt einfache Hilfsprogramme zur Verfügung, mit denen strukturierte Textformate direkt in Lua-Tabellen dekodiert werden können.

`sbomgen.json_decode (str)`

Analysiert eine JSON-Zeichenfolge in eine Lua-Tabelle.

- Gibt zurück: `table|nil, err`

```
local doc, err = sbomgen.json_decode('{"name":"requests","version":"2.28.1"}')
if err then return end
sbomgen.log_info(doc.name)
```

`sbomgen.xml_decode (str)`

Analysiert eine XML-Zeichenfolge in eine Lua-Tabelle.

- Gibt zurück: `table|nil, err`

XML-Werte haben die folgende Form:

- `_name`— Name des Elements
- `_attr`— Attributtabelle, falls vorhanden
- `_text`— getrimmter Textinhalt, falls vorhanden
- numerische Indizes `1..n` — untergeordnete Elemente

```
local doc, err = sbomgen.xml_decode('<package id="Newtonsoft.Json" version="13.0.3" /
>')
if err then return end
sbomgen.log_info(doc._attr.id)
```

Windows Registry

Diese Funktionen bieten schreibgeschützten Zugriff auf die Windows-Registrierung. Gibt bei Artefakten, die nicht zu Windows gehören, `registry_open_key` einen Fehler zurück. Der Registrierungs-Accessor wird bei der ersten Verwendung verzögert initialisiert und unterstützt sowohl Live-Windows-API-Zugriff (Localhost-Scans unter Windows) als auch dateibasiertes REGF-Hive-Parsing (Scans). container/volume

`sbomgen.registry_open_key (Pfad)`

Öffnet einen Registrierungsschlüssel. Gibt ein Schlüssel-Handle zurück, das mit geschlossen werden muss`registry_close`.

- Gibt zurück: `key, err`

```
local key, err = sbomgen.registry_open_key("SOFTWARE\\Microsoft\\Windows\
\CurrentVersion\\Uninstall\\7-Zip")
if err then return end
-- use key...
sbomgen.registry_close(key)
```

`sbomgen.registry_get_string (Schlüssel, Wertname)`

Liest einen Zeichenkettenwert aus einem geöffneten Registrierungsschlüssel.

- Gibt zurück: `string, err`

```
local version, err = sbomgen.registry_get_string(key, "DisplayVersion")
```

`sbomgen.registry_get_integer (Schlüssel, Wertname)`

Liest einen Integer-Wert aus einem geöffneten Registrierungsschlüssel.

- Gibt zurück: `number, err`

sbomgen.registry_get_strings (Schlüssel, Wertname)

Liest einen Wert mit mehreren Zeichenketten (REG_MULTI_SZ) aus einem geöffneten Registrierungsschlüssel. Gibt eine Tabelle mit Zeichenketten zurück.

- Gibt zurück: `{string, ...}, err`

```
local paths, err = sbomgen.registry_get_strings(key, "DependsOnService")
if paths then
    for _, p in ipairs(paths) do
        sbomgen.log_info("depends on: " .. p)
    end
end
```

sbomgen.registry_get_subkeys (Schlüssel)

Gibt alle Unterschlüsselnamen unter einem geöffneten Registrierungsschlüssel zurück.

- Gibt zurück: `{string, ...}, err`

```
local subkeys, err = sbomgen.registry_get_subkeys(key)
for _, name in ipairs(subkeys) do
    local subkey, err = sbomgen.registry_open_key(parent_path .. "\\" .. name)
    -- ...
end
```

sbomgen.registry_close (Schlüssel)

Schließt ein Registrierungsschlüssel-Handle. Schlüsselhandles werden ebenfalls automatisch vom Garbage-Collector geschlossen, es wird jedoch empfohlen, sie explizit zu schließen.

Protokollierung

Protokollnachrichten werden in die Konsolenausgabe von sbomgen geschrieben. Jeder von einem Plugin ausgegebenen Nachricht wird automatisch das Quelllabel und das Ökosystem des Plugins vorangestellt, zum Beispiel:

```
[custom:python-pip] Parsing requirements.txt
```

`log_info`log_warn, und `log_error` immer drucken. `log_debug`druckt nur, wenn sbomgen mit aufgerufen wird. `--verbose`

Funktion	Stufe	Standardmäßig sichtbar?
`sbomgen.log_debug( message)`	DEBUG	Nein — erfordert `--verbose`
`sbomgen.log_info(m essage)`	INFO	Ja
`sbomgen.log_warn(m essage)`	WARN	Ja
`sbomgen.log_error( message)`	ERROR	Ja

`string.format`Für formatierte Nachrichten verwenden:

```
sbomgen.log_info(string.format("found %d packages in %s", count, file_path))
```

Debugging-Funktionen

`sbomgen.breakpoint (Nachricht)`

Druckt `message` auf stderr und blockiert die Ausführung, bis der Benutzer die Eingabetaste drückt. Wenn `message` es weggelassen wird, wird eine Standardnachricht gedruckt.

Verwenden Sie dies als einfachen Debugger, indem Sie Breakpoints an wichtigen Punkten in Ihrem Plugin platzieren und mit ausführen, um die umliegende Protokollausgabe `--verbose` zu sehen.

```
sbomgen.log_info("state: " .. some_variable)
sbomgen.breakpoint("paused after state dump — press Enter to continue")
```

API testen

Funktionen in der globalen `testing` Tabelle sind nur in Plugin-Testdateien (`*_test.lua`) verfügbar, die von geladen wurden`inspector-sbomgen plugin test`. Sie sind zur Laufzeit

in Discovery- oder Collection-Plugins nicht verfügbar. Die vollständige `sbomgen.*` API ist auch in Testdateien verfügbar, aber `sbomgen.*` Funktionen, die beispielsweise `sbomgen.read_file()` ein Artefakt erfordern, führen nur dann zu aussagekräftigen Ergebnissen, wenn sie innerhalb eines Scans aufgerufen werden. Eine ausführliche Anleitung finden Sie in derthe section called “Leitfaden zum Testen von Plugins”.

Funktionen scannen

Jede Scanfunktion erzeugt ein Artefakt der angegebenen Art, führt die Discovery→Collection-Pipeline des aktuellen Plugins anhand dieses Artefakts durch und gibt die resultierenden Ergebnisse zurück. Das `path` Argument wird relativ zum Verzeichnis der Testdatei aufgelöst.

Funktion	Art des Artefakts
`testing.scan_directory(path)`	Verzeichnis
`testing.scan_archive(path)`	Verzeichnis (Alias von`scan_directory`)
`testing.scan_localhost(path)`	Lokaler Host
`testing.scan_binary(path)`	Binär
`testing.scan_volume(path)`	Volume
`testing.scan_container(path)`	Behälter

Alle sechs geben eine Ergebnistabelle mit der folgenden Form zurück.

Form des Ergebnisses

Jede Suchtabelle projiziert nur die unten aufgeführten Felder. Insbesondere `namespace` und `purl_type` werden nicht separat projiziert — sie sind in der gesamten `purl` Zeichenfolge enthalten.

```
local result = testing.scan_directory("_testdata/example")
-- result.findings                          -- array of finding tables
-- result.findings[i].name                  -- string
-- result.findings[i].version               -- string
-- result.findings[i].component_type        -- string
-- result.findings[i].purl                  -- string (the full Package URL, or "" if
 none)
```

```
-- result.findings[i].properties          -- table<string, string>
-- result.findings[i].children            -- array of finding tables (same shape,
 recursive)
```

Assertionen

Funktion	Signature	Beschreibung
`testing.assert_equals`	`(expected: any, actual: any, message?: string)`	Schlägt fehl, wenn`tostring(expected) ~= tostring(actual)` .
`testing.assert_not_equals`	`(expected: any, actual: any, message?: string)`	Schlägt fehl, wenn`tostring(expected) == tostring(actual)` .
`testing.assert_true`	`(value: any, message?: string)`	Schlägt fehl, wenn `value` es `false` oder ist`nil`.
`testing.assert_false`	`(value: any, message?: string)`	Schlägt fehl, wenn `value` nicht `false` und nicht`nil`.
`testing.assert_nil`	`(value: any, message?: string)`	Schlägt fehl`value`, wenn nicht`nil`.
`testing.assert_not_nil`	`(value: any, message?: string)`	Scheitert, wenn `value` es so ist`nil`.
`testing.assert_contains`	`(haystack: string, needle: string, message?: string)`	Schlägt fehl, wenn `haystack` es nicht enthält `needle` (Teilzeichenfolge entspricht).
`testing.assert_matches`	`(str: string, pattern: string, message?: string)`	Schlägt fehl, wenn `str` es nicht mit der angegebenen Go-Regex (RE2) übereinst immt.

Funktion	Signature	Beschreibung
`testing.assert_length`	`(tbl: table, expected: integer, message?: string)`	Schlägt fehl, wenn `#tbl` nicht identisch. `expected`

Kontrollfluss

Funktion	Signature	Beschreibung
`testing.fail`	`(message: string)`	Schlägt den aktuellen Test sofort mit der angegebenen Nachricht fehl.
`testing.skip`	`(message: string)`	Überspringt den aktuellen Test. Das Ergebnis wird als übersprungen und nicht als Fehlschlag gemeldet.

Testen Sie die Entdeckung

Jede globale Lua-Funktion, deren Name `test_` in einem Dateiabgleich mit „beginnt“, `*_test.lua` wird als Test behandelt. Die Testdatei muss neben und `init.lua` in normaler `{phase}/{platform}/{category}/{ecosystem}/` Tiefe liegen. Die Fixture-Daten werden `_testdata/` neben der Testdatei eingegeben — der Runner rutscht nicht hinein, `_testdata/` wenn er nach Testdateien sucht.

Fehlerbehandlung

API-Funktionen, die fehlschlagen können, geben zwei Werte zurück:. `value, err` Bei Erfolg `err` ist`nil`. Bei einem Fehler ist `nil` und `err` ist der erste Wert eine Fehlerzeichenfolge.

```
local content, err = sbomgen.read_file(path)
if err then
    sbomgen.log_error("failed to read " .. path .. ": " .. err)
    return
end
```

```
-- content is safe to use here
```

Wenn ein Plugin einen unbehandelten Lua-Fehler auslöst, protokolliert sbomgen eine Warnung und fährt mit der nächsten Datei oder dem nächsten Plugin fort. Andere Plugins sind nicht betroffen.

Integration von Amazon Inspector-Scans in Ihre CI/CD Pipeline

Die Amazon CI/CD Inspector-Integration verwendet den Amazon Inspector SBOM Generator und die Amazon Inspector Scan API, um Schwachstellenberichte für Container-Images zu erstellen. Der Amazon Inspector SBOM Generator erstellt eine Software-Stückliste (SBOM) für Archive, Container-Images, Verzeichnisse, lokale Systeme sowie kompilierte Dateien und Binärdateien. Go Rust Die Amazon Inspector Scan API scannt die SBOM, um einen Bericht mit Details zu erkannten Sicherheitslücken zu erstellen. Sie können Amazon Inspector Container-Image-Scans in Ihre CI/CD Pipeline integrieren, um nach Software-Schwachstellen zu suchen und Schwachstellenberichte zu erstellen, mit denen Sie Risiken vor der Bereitstellung untersuchen und beheben können. Um Ihre CI/CD Integration einzurichten, können Sie Plugins verwenden oder mit dem Amazon Inspector SBOM Generator und der Amazon Inspector Scan API eine benutzerdefinierte CI/CD Integration erstellen.

Themen

- Plugin-Integration
- Maßgeschneiderte Integration
- Einrichtung eines AWS Konto zur Nutzung der Amazon Inspector CI/CD Inspector-Integration
- Amazon Inspector Dockerfile-Prüfungen
- Erstellen einer benutzerdefinierten CI/CD Pipeline-Integration mit Amazon Inspector Scan
- Verwenden von Amazon Inspector mit GitHub actions
- Den Amazon Inspector verwenden Jenkins Plugin
- Verwenden CodeCatalyst Aktionen mit Amazon Inspector
- Verwenden von Amazon Inspector Scan-Aktionen mit CodePipeline
- Den Amazon Inspector verwenden TeamCity Plugin
- Verwenden von Amazon Inspector mit GitLab components

Plugin-Integration

Amazon Inspector bietet Plugins für unterstützte CI/CD Lösungen. Sie können diese Plugins von ihren jeweiligen Marketplaces aus installieren und sie dann verwenden, um Amazon Inspector Scans als Build-Schritt in Ihre Pipeline aufzunehmen. Im Schritt zur Plugin-Erstellung wird der Amazon

Inspector SBOM-Generator auf dem von Ihnen bereitgestellten Bild ausgeführt und anschließend die Amazon Inspector Scan-API auf der generierten SBOM ausgeführt.

Im Folgenden finden Sie eine Übersicht darüber, wie eine Amazon Inspector CI/CD Inspector-Integration über Plugins funktioniert:

1. Sie konfigurieren eine AWS-Konto , um den Zugriff auf die Amazon Inspector Scan API zu ermöglichen. Detaillierte Anweisungen finden Sie unter Einrichtung eines AWS Konto zur Nutzung der Amazon Inspector CI/CD Inspector-Integration.
2. Sie installieren das Amazon Inspector-Plugin vom Marketplace.
3. Sie installieren und konfigurieren die Amazon Inspector SBOM Generator-Binärdatei. Detaillierte Anweisungen finden Sie unter Amazon Inspector SBOM-Generator.
4. Sie fügen Amazon Inspector Scans als Build-Schritt in Ihre CI/CD Pipeline ein und konfigurieren den Scan.
5. Wenn Sie einen Build ausführen, verwendet das Plugin Ihr Container-Image als Eingabe und führt dann den Amazon Inspector SBOM Generator auf dem Image aus, um eine CycloneDX kompatible SBOM zu generieren.
6. Von dort aus sendet das Plugin die generierte SBOM an einen Amazon Inspector Scan API-Endpunkt, der jede SBOM-Komponente auf Sicherheitslücken überprüft.
7. Die Antwort der Amazon Inspector Scan API wird in einen Schwachstellenbericht in den Formaten CSV, SBOM, JSON und HTML umgewandelt. Der Bericht enthält Details zu allen Sicherheitslücken, die Amazon Inspector gefunden hat.

Unterstützte CI/CD Lösungen

Amazon Inspector unterstützt derzeit die folgenden CI/CD Lösungen. Vollständige Anweisungen zur Einrichtung der CI/CD Integration mithilfe eines Plug-ins erhalten Sie, wenn Sie das Plug-in für Ihre CI/CD Lösung auswählen:

- GitHub Aktionen
- Jenkins-Plugin
- TeamCity-Plugin

Maßgeschneiderte Integration

Wenn Amazon Inspector keine Plug-ins für Ihre CI/CD Lösung bereitstellt, können Sie mithilfe einer Kombination aus dem Amazon Inspector SBOM Generator und der Amazon Inspector Scan API Ihre eigene benutzerdefinierte CI/CD Integration erstellen. Sie können auch eine benutzerdefinierte Integration verwenden, um Scans mithilfe der im Amazon Inspector SBOM Generator verfügbaren Optionen zu optimieren.

Im Folgenden finden Sie einen Überblick darüber, wie eine benutzerdefinierte Amazon Inspector CI/CD Inspector-Integration funktioniert:

1. Sie konfigurieren eine AWS-Konto , um den Zugriff auf die Amazon Inspector Scan API zu ermöglichen. Detaillierte Anweisungen finden Sie unter Einrichtung eines AWS Konto zur Nutzung der Amazon Inspector CI/CD Inspector-Integration.
2. Sie installieren und konfigurieren die Amazon Inspector SBOM Generator-Binärdatei. Detaillierte Anweisungen finden Sie unter Amazon Inspector SBOM-Generator.
3. Sie verwenden den Amazon Inspector SBOM Generator, um eine CycloneDX kompatible SBOM für Ihr Container-Image zu generieren.
4. Sie verwenden die Amazon Inspector Scan API für die generierte SBOM, um einen Schwachstellenbericht zu erstellen.

Anweisungen zum Einrichten einer benutzerdefinierten Integration finden Sie unter. Erstellen einer benutzerdefinierten CI/CD Pipeline-Integration mit Amazon Inspector Scan

Einrichtung eines AWS Konto zur Nutzung der Amazon Inspector CI/CD Inspector-Integration

Um die Amazon Inspector CI/CD Inspector-Integration nutzen zu können, müssen Sie sich für eine registrieren AWS-Konto. Sie AWS-Konto müssen über eine IAM-Rolle verfügen, die Ihrer CI/CD Pipeline Zugriff auf die Amazon Inspector Scan API gewährt. Führen Sie die Aufgaben in den folgenden Themen aus, um sich für eine IAM-Rolle zu registrieren AWS-Konto, einen Administratorbenutzer zu erstellen und eine IAM-Rolle für die Integration zu konfigurieren. CI/CD

> **Note**
>
> Wenn Sie sich bereits für eine angemeldet haben AWS-Konto, können Sie direkt zuKonfigurieren Sie eine IAM-Rolle für die Integration CI/CD.

Themen

- Melde dich für eine an AWS-Konto
- Konfigurieren Sie eine IAM-Rolle für die Integration CI/CD

Melde dich für eine an AWS-Konto

Um loszulegen AWS, benötigen Sie eine AWS-Konto. Es gibt zwei Möglichkeiten, sich anzumelden AWS. Wenn Sie die vollständige Kontrolle über die Kontokonfiguration und den Speicherort Ihrer Inhalte haben möchten oder wenn Sie regulierte Workloads wie HIPAA oder FedRAMP haben, verwenden Sie Sign up for (advanced). AWS Wenn Sie ein neuer Benutzer sind und einen Anmeldenamen verwenden möchten, den Sie bereits besitzen, um sich anzumelden AWS und eine AWS Umgebung mit vorkonfigurierten Standardeinstellungen in select zu erstellen, verwenden Sie Sign up for. AWS-RegionenAWS

Konfigurieren Sie eine IAM-Rolle für die Integration CI/CD

Um Amazon Inspector-Scanning in Ihre CI/CD Pipeline zu integrieren, müssen Sie eine IAM-Richtlinie erstellen, die den Zugriff auf die Amazon Inspector Scan-API ermöglicht, die die Software-Stückliste (SBOMs) scannt. Anschließend können Sie diese Richtlinie einer IAM-Rolle zuordnen, die Ihr Konto für die Ausführung der Amazon Inspector Scan API übernehmen kann.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Richtlinien und wählen Sie dann Richtlinie erstellen aus.
3. Wählen Sie im Policy Editor JSON aus und fügen Sie die folgende Anweisung ein:

   JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "inspector-scan:ScanSbom",
            "Resource": "*"
        }
    ]
}
```

4. Wählen Sie Weiter aus.
5. Geben Sie der Richtlinie beispielsweise `InspectorCICDscan-policy` einen Namen und fügen Sie eine optionale Beschreibung hinzu. Wählen Sie dann Create Policy aus. Diese Richtlinie wird der Rolle angehängt, die Sie in den nächsten Schritten erstellen werden.
6. Wählen Sie im Navigationsbereich der IAM-Konsole Rollen und dann Neue Rolle erstellen aus.
7. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus und fügen Sie die folgende Richtlinie ein:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

8. Wählen Sie Weiter aus.
9. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie zuvor erstellt haben, wählen Sie sie aus und klicken Sie dann auf Weiter.
10. Geben Sie der Rolle beispielsweise `InspectorCICDscan-role` einen Namen und fügen Sie eine optionale Beschreibung hinzu`Create Role`. Wählen Sie dann.

Amazon Inspector Dockerfile-Prüfungen

In diesem Abschnitt wird beschrieben, wie Sie den Amazon Inspector SBOM Generator verwenden, um Bilder zu scannen Dockerfiles und Docker zu speichern, um Fehlkonfigurationen zu erkennen, die zu Sicherheitslücken führen.

Themen

- Verwenden Sbomgen Dockerfile-Prüfungen
- Unterstützte Dockerfile-Prüfungen

Verwenden Sbomgen Dockerfile-Prüfungen

Dockerfile-Prüfungen werden automatisch durchgeführt, wenn eine Datei namens `Dockerfile` oder entdeckt `*.Dockerfile` wird und wenn ein Docker-Image gescannt wird.

Sie können Dockerfile-Prüfungen mit dem Argument deaktivieren. `--skip-scanners dockerfile` Sie können Dockerfile-Prüfungen auch mit allen verfügbaren Scannern kombinieren, z. B. mit Betriebssystemen oder Paketen von Drittanbietern.

Beispiele für Docker-Checkbefehle

Die folgenden Beispielbefehle zeigen, wie SBOMs für Dockerfiles und Docker-Container-Images sowie für Betriebssysteme und Pakete von Drittanbietern generiert werden.

```
# generate SBOM only containing Docker checks for Dockerfiles in a local directory
./inspector-sbomgen directory --path ./project/ --scanners dockerfile

# generate SBOM for container image will by default include Dockerfile checks
./inspector-sbomgen container --image image:tag

# generate SBOM only containing Docker checks for specific Dockerfiles and Alpine,
 Debian, and Rhel OS packages in a local directory
/inspector-sbomgen directory --path ./project/ --scanners dockerfile,dpkg,alpine-
apk,rhel-rpm

# generate SBOM only containing Docker checks for specific Dockerfiles in a local
 directory
./inspector-sbomgen directory --path ./project/ --skip-scanners dockerfile
```

Beispiel für eine Dateikomponente

Im Folgenden finden Sie ein Beispiel für eine Dockerfile-Suche nach einer Dateikomponente.


```
{
     "bom-ref": "comp-2",
      "name": "dockerfile:data/docker/Dockerfile",
      "properties": [
        {
          "name": "amazon:inspector:sbom_scanner:dockerfile_finding:IN-DOCKER-001",
          "value": "affected_lines:27-27"
        }
      ],
      "type": "file"
    },
```


Beispiel einer Komponente zur Reaktion auf Sicherheitslücken

Im Folgenden finden Sie ein Beispiel für eine Dockerfile-Suche nach einer Komponente zur Reaktion auf Sicherheitslücken.


```
{
      "advisories": [
        {
          "url": "https://docs.docker.com/develop/develop-images/instructions/"
        }
      ],
      "affects": [
        {
          "ref": "comp-2"
        }
      ],
      "analysis": {
        "state": "in_triage"
      },
      "bom-ref": "vuln-13",
      "created": "2024-03-27T14:36:39Z",
      "description": "apt-get layer caching: Using apt-get update alone in a RUN
 statement causes caching issues and subsequent apt-get install instructions to fail.",
      "id": "IN-DOCKER-001",
      "ratings": [
        {
          "method": "other",
          "severity": "info",
          "source": {
```

```
            "name": "AMAZON_INSPECTOR",
            "url": "https://aws.amazon.com/inspector/"
          }
        }
      ],
      "source": {
        "name": "AMAZON_INSPECTOR",
        "url": "https://aws.amazon.com/inspector/"
      },
      "updated": "2024-03-27T14:36:39Z"
    },
```

> **Note**
>
> Wenn Sie Sbomgen ohne das `--scan-sbom` Flag aufrufen, können Sie nur Rohergebnisse von Dockerfile anzeigen.

Unterstützte Dockerfile-Prüfungen

SbomgenDockerfile-Prüfungen werden für Folgendes unterstützt:

- Das Sudo-Binärpaket
- APT-Dienstprogramme von Debian
- Hartcodierte Geheimnisse
- Root-Container
- Schwächung der Befehlsflags zur Laufzeit
- Umgebungsvariablen, die die Laufzeit schwächen

Jede dieser Dockerfile-Prüfungen hat einen entsprechenden Schweregrad, der oben in den folgenden Themen angegeben ist.

> **Note**
>
> Die in den folgenden Themen beschriebenen Empfehlungen basieren auf bewährten Verfahren der Branche.

Das Sudo-Binärpaket

> **Note**
>
> Der Schweregrad dieser Prüfung ist Info.

Wir empfehlen, das Sudo-Binärpaket nicht zu installieren oder zu verwenden, da es ein unvorhersehbares TTY- und Signalweiterleitungsverhalten aufweist. Weitere Informationen finden Sie unter Benutzer auf der Docker Docs-Website. Wenn Ihr Anwendungsfall eine ähnliche Funktionalität wie das Sudo-Binärpaket erfordert, empfehlen wir die Verwendung von Gosu.

Debian APT-Dienstprogramme

> **Note**
>
> Der Schweregrad dieser Prüfung ist Hoch.

Im Folgenden finden Sie bewährte Methoden für die Verwendung von Debian APT-Dienstprogrammen.

Kombinieren von **`apt-get-Befehlen`** in einer einzigen **`Run-Anweisung, um Probleme`** beim Zwischenspeichern zu vermeiden

Wir empfehlen, `apt-get` Befehle in einer einzigen RUN-Anweisung innerhalb Ihres Docker-Containers zu kombinieren. Die `apt-get update` alleinige Verwendung führt zu Caching-Problemen und nachfolgenden `apt-get install` Anweisungen schlagen fehl. Weitere Informationen finden Sie unter apt-get auf der Docker Docs-Website.

> **Note**
>
> Das beschriebene Caching-Verhalten kann auch innerhalb Ihres Docker Containers auftreten, wenn die Docker-Container-Software veraltet ist.

Verwenden Sie das APT-Befehlszeilenprogramm auf nicht interaktive Weise

Wir empfehlen, das APT-Befehlszeilenprogramm interaktiv zu verwenden. Das APT-Befehlszeilenprogramm ist als Tool für Endbenutzer konzipiert, und sein Verhalten ändert

sich zwischen den Versionen. Weitere Informationen finden Sie auf der Debian-Website unter Verwendung von Skripten und Unterschiede zu anderen APT-Tools.

Hard-coded Geheimnisse

> **Note**
>
> Der Schweregrad dieser Prüfung ist Kritisch.

Vertrauliche Informationen in Ihrem Dockerfile werden als fest codiertes Geheimnis betrachtet. Die folgenden hartcodierten Geheimnisse können durch Sbomgen Docker-Dateiprüfungen identifiziert werden:

- AWS Zugriffsschlüssel-IDs — `AKIAIOSFODNN7EXAMPLE`
- AWS geheime Schlüssel — `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`
- DockerHub persönliche Zugriffstoken — `dckr_pat_thisisa27charexample1234567`
- GitHub persönliche Zugriffstoken — `ghp_examplev61wY7Pj1YnotrealUoY123456789`
- GitLab persönliche Zugriffstoken — `glpat-12345example12345678`

Root-Container

> **Note**
>
> Der Schweregrad dieser Prüfung ist Info.

Wir empfehlen, Docker-Container ohne Root-Rechte auszuführen. Für containerisierte Workloads, die ohne Root-Rechte nicht ausgeführt werden können, empfehlen wir, Ihre Anwendungen nach einem Prinzip mit den geringsten Rechten zu erstellen. Weitere Informationen finden Sie unter Benutzer auf der Docker Docs-Website.

Umgebungsvariablen, die die Laufzeit schwächen

> **Note**
>
> Der Schweregrad dieser Prüfung ist Hoch.

Verschiedene Befehlszeilenprogramme oder Runtimes in Programmiersprachen unterstützen die Umgehung sicherer Standardeinstellungen, wodurch die Ausführung mit unsicheren Methoden ermöglicht wird.

NODE_TLS_REJECT_UNAUTHORIZED=0

Wenn Node.js Prozesse mit der Einstellung auf ausgeführt werden, ist die TLS-Zertifikatsvalidierung deaktiviert. `NODE_TLS_REJECT_UNAUTHORIZED 0` Weitere Informationen finden Sie auf der Website unter NODE_TLS_REJECT_UNAUTHORIZED=0. Node.js

GIT_SSL_NO_VERIFY=*

Wenn Git-Befehlszeilenprozesse mit `GIT_SSL_NO_VERIFY` set ausgeführt werden, überspringt Git die Überprüfung von TLS-Zertifikaten. Weitere Informationen finden Sie unter Umgebungsvariablen auf der Git-Website.

PIP_TRUSTED_HOST=*

Wenn Python Pip-Befehlszeilenprozesse mit `PIP_TRUSTED_HOST` set ausgeführt werden, überspringt Pip die Überprüfung von TLS-Zertifikaten auf der angegebenen Domain. Weitere Informationen finden Sie unter --trusted-host auf der Pip-Website.

npm_config_strict_ssl=False

Wenn Node.js npm-Befehlszeilenprozesse mit dem `NPM_CONFIG_STRICT_SSL` Wert false ausgeführt werden, stellt das Node Package Manager (npm) -Hilfsprogramm eine Verbindung zur NPM-Registrierung her, ohne die TLS-Zertifikate zu überprüfen. Weitere Informationen finden Sie unter strict-ssl auf der Website von npm Docs.

Befehlsflags werden zur Laufzeit geschwächt

> **Note**
>
> Der Schweregrad dieser Prüfung ist Hoch.

Ähnlich wie bei Umgebungsvariablen, die die Laufzeit schwächen, unterstützen mehrere Befehlszeilenprogramme oder Laufzeitumgebungen in Programmiersprachen die Umgehung sicherer Standardwerte, was die Ausführung mit unsicheren Methoden ermöglicht.

`npm ——strict-ssl=false`

Wenn Node.js npm-Befehlszeilenprozesse mit dem `--strict-ssl=false` Flag ausgeführt werden, stellt das Node Package Manager (npm) -Hilfsprogramm eine Verbindung zur NPM-Registrierung her, ohne TLS-Zertifikate zu validieren. Weitere Informationen finden Sie unter strict-ssl auf der npm Docs-Website.

`apk – zulassen-nicht vertrauenswürdig`

Wenn das Alpine Package Keeper Hilfsprogramm mit dem `--allow-untrusted` Flag ausgeführt wird, `apk` werden Pakete ohne oder ohne vertrauenswürdige Signaturen installiert. Weitere Informationen finden Sie im folgenden Repository auf der Apline-Website.

`apt-get ——allow-unauthenticated`

Wenn das `apt-get` Debian-Paket-Hilfsprogramm mit dem `--allow-unauthenticated` Flag ausgeführt wird, überprüft es nicht die Gültigkeit des Pakets. `apt-get` Weitere Informationen finden Sie unter APT-Get(8) auf der Debian-Webseite.

`pip – trusted-host`

Wenn das Python Pip-Hilfsprogramm mit dem `--trusted-host` Flag ausgeführt wird, umgeht der angegebene Hostname die TLS-Zertifikatsvalidierung. Weitere Informationen finden Sie unter --trusted-host auf der Pip-Website.

`rpm – nodigest, ——nosignature, ——noverify, ——nofiledigest`

Wenn der RPM-based Paketmanager mit den `--nofiledigest` Flags,, und ausgeführt `rpm` wird `--nodigest` `--nosignature--noverify`, überprüft der RPM-Paketmanager bei der Installation eines Pakets keine Paket-Header, Signaturen oder Dateien. Weitere Informationen finden Sie auf der folgenden RPM-Handbuchseite auf der RPM-Website.

`yum-config-manager – setopt=sslverify false`

Wenn der RPM-based Paketmanager mit dem auf False gesetzten `--setopt=sslverify` Flag ausgeführt `yum-config-manager` wird, validiert der YUM-Paketmanager keine TLS-Zertifikate. Weitere Informationen finden Sie auf der folgenden YUM-Handbuchseite auf der Man7-Website.

`yum – nogpgcheck`

Wenn der RPM-based Paketmanager mit dem `--nogpgcheck` Flag ausgeführt `yum` wird, überspringt der YUM-Paketmanager die Überprüfung der GPG-Signaturen auf Paketen. Weitere Informationen finden Sie unter yum (8) auf der Man7-Website.

`curl – unsicher, curl –k`

Wenn `curl` es mit dem `-k` Flag `--insecure` oder ausgeführt wird, ist die TLS-Zertifikatsvalidierung deaktiviert. Standardmäßig wird jede sichere Verbindung, die `curl` hergestellt wird, als sicher verifiziert, bevor die Übertragung stattfindet. Bei dieser Option können `curl` Sie den Bestätigungsschritt überspringen und ohne Überprüfung fortfahren. Weitere Informationen finden Sie auf der folgenden Curl-Handbuchseite auf der Curl-Website.

`wget – kein Check-Zertifikat`

Wenn `wget` es mit dem `--no-check-certificate` Flag ausgeführt wird, ist die TLS-Zertifikatsvalidierung deaktiviert. Weitere Informationen finden Sie auf der folgenden Wget-Handbuchseite auf der GNU-Website.

Beim Entfernen werden Betriebssystem-Paketdatenbanken in Containern überprüft

> **Note**
>
> Der Schweregrad dieser Prüfung ist Info.

Das Entfernen einer Betriebssystem-Paketdatenbank reduziert die Möglichkeit, das gesamte Inventar der Software eines Container-Images zu scannen. Diese Datenbanken sollten während der Container-Build-Schritte intakt bleiben.

Entfernungsprüfungen für eine Betriebssystem-Paketdatenbank werden für die folgenden Paketmanager unterstützt:

Alpine Package Keeper (APK)

Container-Images, die den APK-Paketmanager für installierte Software verwenden, müssen sicherstellen, dass APK-Systemdateien während eines Builds nicht entfernt werden. Weitere Informationen finden Sie in der Dokumentation zu den APK-Manpages-Systemdateien auf der Arch Linux Website.

Debian-Paketmanager (DPKG)

Container, die den DPKG-Paketmanager verwenden, wie z. B. Debian-, Ubuntu- oder Distroless-basierte Images, müssen sicherstellen, dass die DPKG-Datenbank während eines Container-Builds nicht entfernt wird. Weitere Informationen finden Sie in der Dokumentation zu den DPKG-Manpages-Systemdateien auf der Website. Ubuntu

RPM-Paketmanager (RPM)

Container, die den RPM Package Manager (yum/dnf) verwenden, wie Amazon Linux oder Red Hat Enterprise Linux, müssen sicherstellen, dass die RPM-Datenbank während eines Container-Builds nicht entfernt wird. Weitere Informationen finden Sie in der Dokumentation zu den RPM-Manpages-Systemdateien auf der RPM-Website.

Erstellen einer benutzerdefinierten CI/CD Pipeline-Integration mit Amazon Inspector Scan

Wir empfehlen Ihnen, die Amazon CI/CD Inspector-Plugins zu verwenden, wenn die Amazon CI/CD Inspector-Plugins für Ihre CI/CD Lösung verfügbar sind. Wenn die Amazon CI/CD Inspector-Plugins für Ihre CI/CD Lösung nicht verfügbar sind, können Sie eine Kombination aus dem Amazon Inspector SBOM Generator und der Amazon Inspector Scan API verwenden, um eine benutzerdefinierte CI/CD Integration zu erstellen. In den folgenden Schritten wird beschrieben, wie Sie eine benutzerdefinierte CI/CD Pipeline-Integration mit Amazon Inspector Scan erstellen.

> **Tip**
>
> Sie können den Amazon Inspector SBOM Generator (Sbomgen) verwenden, um Schritt 3 und Schritt 4 zu überspringen, wenn Sie Ihre SBOM mit einem einzigen Befehl generieren und scannen möchten.

Schritt 1. Konfigurieren AWS-Konto

Konfigurieren Sie einen AWS-Konto , der Zugriff auf die Amazon Inspector Scan API bietet. Weitere Informationen finden Sie unter Einrichtung eines AWS Konto zur Nutzung der Amazon Inspector CI/CD Inspector-Integration.

Schritt 2. Installation Sbomgen Binary

Installieren und konfigurieren Sie die Sbomgen Binärdatei. Weitere Informationen finden Sie unter Installieren der Sbomgen.

Schritt 3. Verwenden Sbomgen

Verwenden Sie denSbomgen, um eine SBOM-Datei für ein Container-Image zu erstellen, das Sie scannen möchten.

Sie können das folgende Beispiel verwenden. *image:id*Ersetzen Sie es durch den Namen des Bilds, das Sie scannen möchten. *sbom_path.json*Ersetzen Sie durch den Speicherort, an dem Sie die SBOM-Ausgabe speichern möchten.

Beispiel

```
./inspector-sbomgen container --image image:id -o sbom_path.json
```

Schritt 4. Aufrufen der Amazon Inspector Scan API

Rufen Sie die `inspector-scan` API auf, um die generierte SBOM zu scannen und einen Schwachstellenbericht zu erstellen.

Sie können das folgende Beispiel verwenden. *sbom_path.json*Ersetzen Sie durch den Speicherort einer gültigen CyclonedX-kompatiblen SBOM-Datei. Ersetzen Sie es *ENDPOINT* durch den API-Endpunkt für den Ort, an AWS-Region dem Sie derzeit authentifiziert sind. Ersetzen Sie es *REGION* durch die entsprechende Region.

Beispiel

```
aws inspector-scan scan-sbom --sbom file://sbom_path.json --endpoint
ENDPOINT-URL --region REGION
```

Eine vollständige Liste der AWS-Regionen Endpunkte finden Sie unter Regionen und Endpunkte.

(Optional) Schritt 5. Generieren und scannen Sie SBOM mit einem einzigen Befehl

> **Note**
>
> Führen Sie diesen Schritt nur aus, wenn Sie Schritt 3 und Schritt 4 übersprungen haben.

Generieren und scannen Sie Ihre SBOM mit einem einzigen Befehl mithilfe der Markierung. `--scan-bom`

Sie können das folgende Beispiel verwenden. *image:id*Ersetzen Sie es durch den Namen des Bilds, das Sie scannen möchten. *profile*Durch das entsprechende Profil ersetzen. *REGION*Durch die entsprechende Region ersetzen. */tmp/scan.json*Ersetzen Sie durch den Speicherort der Datei scan.json im Verzeichnis tmp.

Beispiel

```
./inspector-sbomgen container --image image:id --scan-sbom --aws-profile
profile --aws-region REGION -o /tmp/scan.json
```

Eine vollständige Liste der Endpunkte finden Sie unter Regionen AWS-Regionen und Endpunkte.

API-Ausgabeformate

Die Amazon Inspector Scan API kann einen Schwachstellenbericht im CycloneDX 1.5-Format oder Amazon Inspector Finding JSON ausgeben. Die Standardeinstellung kann mithilfe des `--output-format` Flags geändert werden.

Beispiel für Cyclone DX Ausgabe im 1.5-Format — Linux


```
{
  "status": "SBOM parsed successfully, 1 vulnerabilities found",
  "sbom": {
    "bomFormat": "CycloneDX",
    "specVersion": "1.5",
    "serialNumber": "urn:uuid:0077b45b-ff1e-4dbb-8950-ded11d8242b1",
    "metadata": {
      "properties": [
        {
          "name": "amazon:inspector:sbom_scanner:critical_vulnerabilities",
          "value": "1"
        },
        {
          "name": "amazon:inspector:sbom_scanner:high_vulnerabilities",
          "value": "0"
        },
        {
          "name": "amazon:inspector:sbom_scanner:medium_vulnerabilities",
          "value": "0"
        },
        {
          "name": "amazon:inspector:sbom_scanner:low_vulnerabilities",
          "value": "0"
        }
      ],
      "tools": [
        {
          "name": "CycloneDX SBOM API",
```

```
          "vendor": "Amazon Inspector",
          "version": "empty:083c9b00:083c9b00:083c9b00"
        }
      ],
      "timestamp": "2023-06-28T14:15:53.760Z"
    },
    "components": [
      {
        "bom-ref": "comp-1",
        "type": "library",
        "name": "log4j-core",
        "purl": "pkg:maven/org.apache.logging.log4j/log4j-core@2.12.1",
        "properties": [
          {
            "name": "amazon:inspector:sbom_scanner:path",
            "value": "/home/dev/foo.jar"
          }
        ]
      }
    ],
    "vulnerabilities": [
      {
        "bom-ref": "vuln-1",
        "id": "CVE-2021-44228",
        "source": {
          "name": "NVD",
          "url": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228"
        },
        "references": [
          {
            "id": "GHSA-jfh8-c2jp-5v3q",
            "source": {
              "name": "GITHUB",
              "url": "https://github.com/advisories/GHSA-jfh8-c2jp-5v3q"
            }
          }
        ],
        "ratings": [
          {
            "source": {
              "name": "NVD",
              "url": "https://www.first.org/cvss/v3-1/"
            },
            "score": 10.0,
```

```
            "severity": "critical",
            "method": "CVSSv31",
            "vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H"
          },
          {
            "source": {
              "name": "NVD",
              "url": "https://www.first.org/cvss/v2/"
            },
            "score": 9.3,
            "severity": "critical",
            "method": "CVSSv2",
            "vector": "AC:M/Au:N/C:C/I:C/A:C"
          },
          {
            "source": {
              "name": "EPSS",
              "url": "https://www.first.org/epss/"
            },
            "score": 0.97565,
            "severity": "none",
            "method": "other",
            "vector": "model:v2023.03.01,date:2023-06-27T00:00:00+0000"
          },
          {
            "source": {
              "name": "GITHUB",
              "url": "https://github.com/advisories/GHSA-jfh8-c2jp-5v3q"
            },
            "score": 10.0,
            "severity": "critical",
            "method": "CVSSv31",
            "vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H"
          }
        ],
        "cwes": [
          400,
          20,
          502
        ],
        "description": "Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security
releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages,
and parameters do not protect against attacker controlled LDAP and other JNDI related
endpoints. An attacker who can control log messages or log message parameters can
```

```
 execute arbitrary code loaded from LDAP servers when message lookup substitution is
 enabled. From log4j 2.15.0, this behavior has been disabled by default. From version
 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely
 removed. Note that this vulnerability is specific to log4j-core and does not affect
 log4net, log4cxx, or other Apache Logging Services projects.",
        "advisories": [
          {
            "url": "https://www.intel.com/content/www/us/en/security-center/advisory/
intel-sa-00646.html"
          },
          {
            "url": "https://support.apple.com/kb/HT213189"
          },
          {
            "url": "https://msrc-blog.microsoft.com/2021/12/11/microsofts-response-to-
cve-2021-44228-apache-log4j2/"
          },
          {
            "url": "https://logging.apache.org/log4j/2.x/security.html"
          },
          {
            "url": "https://www.debian.org/security/2021/dsa-5020"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-479842.pdf"
          },
          {
            "url": "https://www.oracle.com/security-alerts/alert-cve-2021-44228.html"
          },
          {
            "url": "https://www.oracle.com/security-alerts/cpujan2022.html"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-714170.pdf"
          },
          {
            "url": "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/M5CSVUNV4HWZZXGOKNSK6L7RPM7BOKIB/"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-661247.pdf"
```

```
          },
          {
            "url": "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/VU57UJDCFIASIO35GC55JMKSRXJMCDFM/"
          },
          {
            "url": "https://www.oracle.com/security-alerts/cpuapr2022.html"
          },
          {
            "url": "https://twitter.com/kurtseifried/status/1469345530182455296"
          },
          {
            "url": "https://tools.cisco.com/security/center/content/
CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd"
          },
          {
            "url": "https://lists.debian.org/debian-lts-announce/2021/12/msg00007.html"
          },
          {
            "url": "https://www.kb.cert.org/vuls/id/930724"
          }
        ],
        "created": "2021-12-10T10:15:00Z",
        "updated": "2023-04-03T20:15:00Z",
        "affects": [
          {
            "ref": "comp-1"
          }
        ],
        "properties": [
          {
            "name": "amazon:inspector:sbom_scanner:exploit_available",
            "value": "true"
          },
          {
            "name": "amazon:inspector:sbom_scanner:exploit_last_seen_in_public",
            "value": "2023-03-06T00:00:00Z"
          },
          {
            "name": "amazon:inspector:sbom_scanner:cisa_kev_date_added",
            "value": "2021-12-10T00:00:00Z"
          },
          {
            "name": "amazon:inspector:sbom_scanner:cisa_kev_date_due",
```

```
            "value": "2021-12-24T00:00:00Z"
          },
          {
            "name": "amazon:inspector:sbom_scanner:fixed_version:comp-1",
            "value": "2.15.0"
          }
        ]
      }
    ]
  }
}
```

Beispiel für Cyclone DX Ausgabe im 1.5-Format — Windows

```
{
  "sbom": {
    "specVersion": "1.5",
    "metadata": {
      "tools": {
        "services": [
          {
            "name": "Amazon Inspector Scan SBOM API",
            "version": "d79c681c+d73b8663+5e50a5ab"
          }
        ]
      },
      "properties": [
        {
          "name": "amazon:inspector:sbom_scanner:critical_vulnerabilities",
          "value": "0"
        },
        {
          "name": "amazon:inspector:sbom_scanner:high_vulnerabilities",
          "value": "0"
        },
        {
          "name": "amazon:inspector:sbom_scanner:medium_vulnerabilities",
          "value": "1"
        },
        {
          "name": "amazon:inspector:sbom_scanner:low_vulnerabilities",
          "value": "0"
        },
```

```
        {
          "name": "amazon:inspector:sbom_scanner:other_vulnerabilities",
          "value": "0"
        }
      ],
      "timestamp": "2026-03-17T00:00:52.344Z"
    },
    "components": [
      {
        "bom-ref": "comp-1",
        "name": "defender",
        "purl": "pkg:generic/microsoft/defender@4.18.25110.5",
        "type": "application",
        "version": "4.18.25110.5",
        "properties": [
          {
            "name": "amazon:inspector:sbom_scanner:source_file_scanner",
            "value": "windows-apps"
          },
          {
            "name": "amazon:inspector:sbom_scanner:source_package_collector",
            "value": "windows-app-defender"
          },
          {
            "name": "amazon:inspector:sbom_scanner:path",
            "value": "vol-0d994b0984fdaa2af:\\ProgramData\\Microsoft\\Windows Defender\
\platform\\4.18.25110.5-0"
          }
        ]
      }
    ],
    "serialNumber": "urn:uuid:6bed582d-191e-4cb7-9875-950dd0b99700",
    "bomFormat": "CycloneDX",
    "vulnerabilities": [
      {
        "advisories": [
          {
            "url": "https://support.microsoft.com/help/5011487"
          },
          {
            "url": "https://catalog.update.microsoft.com/v7/site/Search.aspx?
q=KB5011487"
          }
        ],
```

```
        "bom-ref": "vuln-1",
        "references": [
          {
            "id": "CVE-2022-23278",
            "source": {
              "name": "MICROSOFT",
              "url": "https://msrc.microsoft.com/update-guide/vulnerability/
CVE-2022-23278"
            }
          }
        ],
        "ratings": [
          {
            "severity": "none",
            "score": 0.02691,
            "method": "other",
            "vector": "model:v2025.03.14,date:2026-03-15T12:55:00Z",
            "source": {
              "name": "EPSS",
              "url": "https://api.first.org/data/v1/epss?cve=CVE-2022-23278"
            }
          },
          {
            "severity": "medium",
            "score": 5.9,
            "method": "CVSSv31",
            "vector": "CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N",
            "source": {
              "name": "MICROSOFT",
              "url": "https://catalog.update.microsoft.com/v7/site/Search.aspx?
q=KB5011487"
            }
          }
        ],
        "created": "2022-03-08T08:00:00Z",
        "description": "Security Update for Defender (2022-03). Install KB5011487 to
 remediate. A reboot is required for this update to take effect.",
        "affects": [
          {
            "ref": "comp-1"
          }
        ],
        "id": "KB5011487",
        "source": {
```

```
          "name": "MICROSOFT",
          "url": "https://catalog.update.microsoft.com/v7/site/Search.aspx?q=KB5011487"
        },
        "published": "2022-03-08T08:00:00Z",
        "analysis": {
          "state": "in_triage"
        },
        "properties": [
          {
            "name": "amazon:inspector:sbom_scanner:priority",
            "value": "standard"
          },
          {
            "name": "amazon:inspector:sbom_scanner:priority_intelligence",
            "value": "unverified"
          },
          {
            "name": "amazon:inspector:sbom_scanner:fixed_version:comp-1",
            "value": "10.0.19042.1586"
          }
        ]
      }
    ]
  }
}
```

Beispiel für eine Ausgabe im Inspector-Format — Linux

```
{
  "status": "SBOM parsed successfully, 1 vulnerability found",
  "inspector": {
    "messages": [
      {
        "name": "foo",
        "purl": "pkg:maven/foo@1.0.0", // Will not exist in output if missing in sbom
        "info": "Component skipped: no rules found."
      }
    ],
    "vulnerability_count": {
      "critical": 1,
      "high": 0,
      "medium": 0,
      "low": 0
```

```
    },
    "vulnerabilities": [
      {
        "id": "CVE-2021-44228",
        "severity": "critical",
        "source": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228",
        "related": [
          "GHSA-jfh8-c2jp-5v3q"
        ],
        "description": "Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security
 releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages,
 and parameters do not protect against attacker controlled LDAP and other JNDI related
 endpoints. An attacker who can control log messages or log message parameters can
 execute arbitrary code loaded from LDAP servers when message lookup substitution is
 enabled. From log4j 2.15.0, this behavior has been disabled by default. From version
 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely
 removed. Note that this vulnerability is specific to log4j-core and does not affect
 log4net, log4cxx, or other Apache Logging Services projects.",
        "references": [
          "https://www.intel.com/content/www/us/en/security-center/advisory/intel-
sa-00646.html",
          "https://support.apple.com/kb/HT213189",
          "https://msrc-blog.microsoft.com/2021/12/11/microsofts-response-to-
cve-2021-44228-apache-log4j2/",
          "https://logging.apache.org/log4j/2.x/security.html",
          "https://www.debian.org/security/2021/dsa-5020",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-479842.pdf",
          "https://www.oracle.com/security-alerts/alert-cve-2021-44228.html",
          "https://www.oracle.com/security-alerts/cpujan2022.html",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-714170.pdf",
          "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/M5CSVUNV4HWZZXGOKNSK6L7RPM7BOKIB/",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-661247.pdf",
          "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/VU57UJDCFIASIO35GC55JMKSRXJMCDFM/",
          "https://www.oracle.com/security-alerts/cpuapr2022.html",
          "https://twitter.com/kurtseifried/status/1469345530182455296",
          "https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-
sa-apache-log4j-qRuKNEbd",
          "https://lists.debian.org/debian-lts-announce/2021/12/msg00007.html",
          "https://www.kb.cert.org/vuls/id/930724"
        ],
        "created": "2021-12-10T10:15:00Z",
```

```
        "updated": "2023-04-03T20:15:00Z",
        "properties": {
          "cisa_kev_date_added": "2021-12-10T00:00:00Z",
          "cisa_kev_date_due": "2021-12-24T00:00:00Z",
          "cwes": [
            400,
            20,
            502
          ],
          "cvss": [
            {
              "source": "NVD",
              "severity": "critical",
              "cvss3_base_score": 10.0,
              "cvss3_base_vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H",
              "cvss2_base_score": 9.3,
              "cvss2_base_vector": "AC:M/Au:N/C:C/I:C/A:C"
            },
            {
              "source": "GITHUB",
              "severity": "critical",
              "cvss3_base_score": 10.0,
              "cvss3_base_vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H"
            }
          ],
          "epss": 0.97565,
          "exploit_available": true,
          "exploit_last_seen_in_public": "2023-03-06T00:00:00Z"
        },
        "affects": [
          {
            "installed_version": "pkg:maven/org.apache.logging.log4j/log4j-
core@2.12.1",
            "fixed_version": "2.15.0",
            "path": "/home/dev/foo.jar"
          }
        ]
      }
    ]
  }
}
```

Beispiel für eine Ausgabe im Inspector-Format — Windows


```
{
  "sbom": {
    "vulnerabilities": [
      {
        "severity": "medium",
        "priority_intelligence": "unverified",
        "related": [
          "CVE-2022-23278"
        ],
        "references": [
          "https://support.microsoft.com/help/5011487",
          "https://catalog.update.microsoft.com/v7/site/Search.aspx?q=KB5011487"
        ],
        "created": "2022-03-08T08:00:00Z",
        "description": "Security Update for Defender (2022-03). Install KB5011487 to
 remediate. A reboot is required for this update to take effect.",
        "affects": [
          {
            "path": "vol-0d994b0984fdaa2af:\\ProgramData\\Microsoft\\Windows Defender\
\platform\\4.18.25110.5-0",
            "fixed_version": "10.0.19042.1586",
            "installed_version": "pkg:generic/microsoft/defender@4.18.25110.5"
          }
        ],
        "id": "KB5011487",
        "source": "https://catalog.update.microsoft.com/v7/site/Search.aspx?
q=KB5011487",
        "published": "2022-03-08T08:00:00Z",
        "priority": "standard",
        "properties": {
          "epss": 0.0269099995,
          "cvss": [
            {
              "severity": "medium",
              "cvss_3_base_score": 5.9000000954,
              "cvss_3_base_vector": "CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N",
              "source": "MICROSOFT",
              "url": "https://catalog.update.microsoft.com/v7/site/Search.aspx?
q=KB5011487"
            }
          ]
        }
```

```
      }
    ],
    "vulnerability_count": {
      "high": 0,
      "other": 0,
      "critical": 0,
      "low": 0,
      "medium": 1
    }
  }
}
```

Verwenden von Amazon Inspector mit GitHub actions

Sie können Amazon Inspector mit verwenden GitHub actions, um Amazon Inspector Inspector-Schwachstellenscans zu Ihren GitHub Workflows hinzuzufügen. Dabei werden der Amazon Inspector SBOM Generator und die Amazon Inspector Scan API genutzt, um am Ende Ihres Builds detaillierte Berichte zu erstellen, sodass Sie Risiken vor der Bereitstellung untersuchen und beheben können. Amazon Inspector Vulnerability Scans können je nach Anzahl und Schweregrad der erkannten Sicherheitslücken so konfiguriert werden, dass Workflows erfolgreich oder nicht bestanden werden. Sie können die neueste Version der Amazon Inspector Inspector-Aktion auf der GitHubWebsite einsehen. Informationen zur Integration von Amazon Inspector Scan in Ihre CI/CD Pipeline finden Sie unter Amazon Inspector-Scans in Ihre CI/CD Pipeline integrieren. Eine Liste der Betriebssysteme und Programmiersprachen, die Amazon Inspector unterstützt, finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen.

Den Amazon Inspector verwenden Jenkins Plugin

Das Amazon Jenkins Inspector-Plugin fügt Ihren Jenkins Builds Amazon Inspector Inspector-Schwachstellenscans hinzu. Das Plugin führt den Amazon Inspector SBOM Generator aus, um eine Software-Stückliste (SBOM) für ein Container-Image zu erstellen. Es sendet die SBOM an die Amazon Inspector Scan API und erstellt am Ende Ihres Builds detaillierte Berichte, sodass Sie Risiken vor der Bereitstellung untersuchen und beheben können. Sie können den Build aufgrund der Anzahl und des Schweregrads der Sicherheitslücken, der Ergebnisse des Exploit Prediction Scoring System (EPSS), bestimmter allgemeiner Sicherheitslücken und Exposures (CVEs) oder bösartiger Pakete als fehlschlagen. Die neueste Version des Plugins finden Sie Amazon Inspector Scanneraf der Plugin-Website. Jenkins In den folgenden Schritten wird beschrieben, wie Sie das Amazon Jenkins Inspector-Plugin einrichten.

> **Important**
>
> Das Plugin benötigt Jenkins 2.541.3 oder höher. Der Amazon Inspector SBOM Generator ist nur für Linux verfügbar. Führen Sie Ihre Builds also Linux mit einer x86_64- (amd64) - oder arm64-CPU aus. Bei der automatischen Installationsmethode muss der Controller ebenfalls ausgeführt werden. Jenkins Linux

Schritt 1. Richten Sie eine ein AWS-Konto

Konfigurieren Sie eine AWS-Konto mit einer IAM-Rolle, die den Zugriff auf die Amazon Inspector Scan API ermöglicht. Detaillierte Anweisungen finden Sie unter Einrichtung eines AWS Konto zur Nutzung der Amazon Inspector CI/CD Inspector-Integration.

Schritt 2. Installieren Sie den Amazon Inspector Jenkins Plugin

Das folgende Verfahren beschreibt, wie Sie das Amazon Jenkins Inspector-Plugin vom Jenkins Dashboard aus installieren.

1. Wählen Sie im Jenkins Dashboard Manage Jenkins und dann Plugins aus.
2. Wählen Sie Verfügbare Plugins aus.
3. Suchen Sie nach Amazon Inspector Scanner, wählen Sie ihn aus und klicken Sie dann auf Installieren.

(Optional) Schritt 3. Addition Docker Anmeldeinformationen für Jenkins

> **Note**
>
> Fügen Sie nur Docker Anmeldeinformationen hinzu, wenn sich das Bild in einem privaten Repository befindet. Andernfalls überspringen Sie diesen Schritt.

Das folgende Verfahren beschreibt, wie Sie Docker Anmeldeinformationen vom Jenkins Dashboard Jenkins aus hinzufügen.

1. Wählen Sie im Jenkins Dashboard Manage Jenkins, Credentials und dann System aus.
2. Wählen Sie Global und dann Anmeldeinformationen hinzufügen aus.

3. Wählen Sie Benutzername mit Passwort und dann Weiter aus.
4. Wählen Sie als Bereich Global (Jenkins, Knoten, Elemente, alle untergeordneten Elemente usw.) aus.
5. Geben Sie den Benutzernamen und das Passwort für Ihre Registrierung ein und wählen Sie dann Create. Wenn Ihre Registrierung Zugriffstoken unterstützt, empfehlen wir Ihnen, anstelle Ihres Kontopassworts ein schreibgeschütztes Token als Passwort zu verwenden.

(Optional) Schritt 4. Addition AWS Anmeldeinformationen für Jenkins

Das Plugin ruft die Amazon Inspector Scan API vom Jenkins Controller aus auf. Wenn Sie den Build-Schritt in Schritt 6 konfigurieren, wählen Sie eine der folgenden Methoden für die Authentifizierung des Plug-ins:

- (Empfohlen) IAM-Rolle — Das Plugin nimmt die Rolle an, die Sie in Schritt 1 erstellt haben. Um die Rolle zu übernehmen, verwendet das Plugin die Anmeldeinformationen, über die der Jenkins Controller bereits verfügt, z. B. ein Amazon EC2 EC2-Instance-Profil. Für diese Anmeldeinformationen ist die Genehmigung erforderlich, um die Rolle aufrufen `sts:AssumeRole` zu können. Das Plugin kann auch die Rolle übernehmen, wenn ein OpenID Connect (OIDC) ID-Token ausgegeben wird. Jenkins In OIDC müssen Sie keine langfristigen Anmeldeinformationen speichern. AWS Jenkins
- AWS Anmeldeinformationen — Ein Zugriffsschlüssel für einen IAM-Benutzer, gespeichert in. Jenkins
- AWS Profilname — Ein benanntes Profil in den AWS Konfigurationsdateien des Benutzers, der den Jenkins Controller ausführt.

Wenn Sie nichts davon konfigurieren, verwendet das Plugin die standardmäßige Anmeldeinformationsanbieterkette auf dem Jenkins Controller, z. B. Umgebungsvariablen oder ein Amazon EC2 EC2-Instance-Profil. Jeder Job, der den Build-Schritt verwendet, kann diese Anmeldeinformationen verwenden. Geben Sie ihnen also nur die Berechtigungen der Rolle, die Sie in Schritt 1 erstellt haben. Sie müssen nur Anmeldeinformationen hinzufügen, Jenkins um einen Zugriffsschlüssel oder OIDC zu verwenden. Andernfalls überspringen Sie diesen Schritt.

> **Warning**
>
> Dieses Szenario erfordert IAM-Benutzer mit programmatischem Zugriff und langfristigen Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren,

empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden. Die Zugriffsschlüssel können bei Bedarf aktualisiert werden. Weitere Informationen finden Sie unter Zugriffsschlüssel aktualisieren im IAM-Benutzerhandbuch.

Um einen Zugriffsschlüssel hinzuzufügen

1. Wählen Sie im Jenkins Dashboard Manage Jenkins, Credentials und dann System aus.
2. Wählen Sie Global und dann Anmeldeinformationen hinzufügen aus.
3. Wählen Sie AWS-Anmeldeinformationen und dann Weiter aus.
4. Geben Sie Ihre Zugriffsschlüssel-ID und Ihren geheimen Zugriffsschlüssel ein und wählen Sie dann Create aus.

Um einen OIDC-Berechtigungsnachweis hinzuzufügen

1. Wählen Sie im Jenkins Dashboard Manage Jenkins, Credentials und dann System aus.
2. Wählen Sie Global und dann Anmeldeinformationen hinzufügen aus.
3. Wählen Sie OpenID Connect ID-Token und dann Weiter.
4. Geben Sie `sts.amazonaws.com` für Audience den Text ein und wählen Sie dann Create aus. Um die Aussteller-URI der Anmeldeinformationen zu sehen, öffnen Sie sie und wählen Sie Anmeldeinformationen aktualisieren aus.
5. Erstellen Sie in IAM einen OIDC-Identitätsanbieter mit der Aussteller-URI als Anbieter-URL und als Zielgruppe. `sts.amazonaws.com` Weitere Informationen finden Sie unter Erstellen eines OpenID Connect (OIDC) -Identitätsanbieters in IAM.
6. Fügen Sie der Vertrauensrichtlinie der Rolle, die Sie in Schritt 1 erstellt haben, eine Erklärung hinzu, die es dem Anbieter ermöglicht, Anrufe zu tätigen. `sts:AssumeRoleWithWebIdentity` Fügen Sie dem `aud` Schlüssel des Anbieters eine Bedingung mit dem Wert hinzu`sts.amazonaws.com`. Damit nur bestimmte Jobs die Rolle übernehmen können, fügen Sie dem `sub` Schlüssel außerdem eine Bedingung hinzu, die standardmäßig die URL des Auftrags enthält. Mit der folgenden Beispielanweisung kann nur der Job *`my-job`* auf einem Jenkins Controller unter *`jenkins.example.com`* die Rolle übernehmen.

```
{
    "Effect": "Allow",
    "Principal": {
```

```
        "Federated": "arn:aws:iam::111122223333:oidc-provider/jenkins.example.com/
oidc"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
        "StringEquals": {
            "jenkins.example.com/oidc:aud": "sts.amazonaws.com",
            "jenkins.example.com/oidc:sub": "https://jenkins.example.com/job/my-
job/"
        }
    }
}
```

> **Note**
>
> AWS muss in der Lage sein, den Aussteller-URI über HTTPS zu erreichen, und der URI sollte keine Portnummer enthalten. Wenn Ihr Jenkins Controller über das Internet nicht erreichbar ist oder seine URL einen Port enthält, verwenden Sie einen anderen Aussteller. Weitere Informationen findest du unter Einen Emittenten auswählen in der OpenID Connect Provider Plugin-Dokumentation auf der GitHub Website.

Schritt 5. Sobald Sie die Details auf dieser Seite überprüft haben, klicken Sie auf Jenkins um den HTML-Bericht anzuzeigen

Das Plugin erstellt einen HTML-Bericht, der Inline-CSS und verwendet JavaScript. Standardmäßig werden Jenkins Build-Artefakte mit einer Content Security Policy (CSP) bereitgestellt, die beide blockiert. Daher hat der Bericht kein Design und zeigt keine Ergebnisse an, selbst wenn bei der Suche Sicherheitslücken gefunden wurden. Verwenden Sie eine der folgenden Optionen, damit der Bericht korrekt angezeigt wird.

- (Empfohlen) Build-Artefakte aus einer anderen Domain bereitstellen — Wählen Sie Manage Jenkins und dann System aus. Geben Sie unter Ressourcendateien aus einer anderen Domain bereitstellen eine Resource Root-URL ein, die auf Ihren Jenkins Controller mit einem anderen Hostnamen als Ihrer Jenkins URL verweist. Jenkinswendet den CSP nicht auf Dateien an, die er über diese URL bereitstellt. Weitere Informationen finden Sie in der Jenkins Dokumentation unter Resource Root URL.

- CSP entfernen — Legen Sie für die `hudson.model.DirectoryBrowserSupport.CSP` Systemeigenschaft einen leeren Wert fest. Um die Änderung sofort zu übernehmen, wählen Sie Manage Jenkins und dann Script Console`System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")`, führen Sie den Bericht aus und laden Sie ihn erneut. Nur Jenkins Administratoren können die Skriptkonsole verwenden. Diese Änderung gilt bis zum Jenkins Neustart. Um sie beizubehalten, beginnen Sie Jenkins mit der Java-Option `-Dhudson.model.DirectoryBrowserSupport.CSP=`.

> **Warning**
>
> Durch das Entfernen des CSP kann jede HTML-Datei in einem Workspace oder Build-Artefakt Skripts in den Browsern Ihrer Jenkins Benutzer ausführen. Entfernen Sie es nur, wenn Sie allen Personen vertrauen, die Dateien in Ihren Builds ändern können, und all Ihren Build-Agents. Weitere Informationen finden Sie in der Jenkins Dokumentation unter Configuring Content Security Policy.

Schritt 6: Fügen Sie Amazon Inspector Scan zu Ihrem Build hinzu

Sie können Amazon Inspector Scan als Build-Schritt in einem Freestyle-Projekt oder als Schritt in einer Jenkins Pipeline hinzufügen.

Amazon Inspector Scan als Build-Schritt hinzufügen

1. Wählen Sie auf der Konfigurationsseite Ihres Projekts unter Build Steps die Option Build Step hinzufügen und dann Amazon Inspector Scan aus.
2. Wählen Sie als Inspector-sbomgen Installationsmethode eine der folgenden Optionen:

   - Automatisch — Das Plugin lädt zu Beginn jedes Builds die neueste Version des Amazon Inspector SBOM Generators herunter. Der Jenkins Controller muss laufen Linux und eine Verbindung `amazon-inspector-sbomgen.s3.amazonaws.com` über HTTPS herstellen können.
   - Manuell — Geben Sie den vollständigen Pfad zu einer `inspector-sbomgen` Binärdatei ein, die Sie heruntergeladen haben, z. `/opt/inspector/inspector-sbomgen` B. Wählen Sie diese Option, um eine bestimmte Version des Amazon Inspector SBOM Generators zu verwenden oder wenn der Jenkins Controller sie nicht herunterladen kann. Das Plugin

überprüft den Pfad auf dem Jenkins Controller, führt die Binärdatei jedoch auf dem Computer aus, auf dem der Build ausgeführt wird. Wenn es sich um unterschiedliche Maschinen handelt, platzieren Sie die Binärdatei auf beiden in demselben Pfad. Die Downloadlinks finden Sie unter Installation Sbomgen.

3. Geben Sie als Bild-ID das zu scannende Bild ein. Das Bild kann lokal, remote oder archiviert sein, und sein Name muss der Docker Benennungskonvention entsprechen:

   - Für ein lokales oder ein Remote-Image: `NAME[:TAG|@DIGEST]`
   - Für ein in eine TAR-Datei exportiertes Bild: `/path/to/image.tar`

4. (Optional) Geben Sie unter Report Artifact Name ein Präfix für den Dateinamen der SBOM ein, die der Build archiviert. Der Standardwert ist `default-report`. Der Name kann bis zu 255 Buchstaben, Zahlen, Punkte (.), Unterstriche (_) und Bindestriche (-) enthalten.
5. (Optional) Geben Sie unter Dateien überspringen die Dateien und Verzeichnisse ein, die vom Scan ausgeschlossen werden sollen, entweder eine pro Zeile oder getrennt durch Kommas ohne Leerzeichen. Ziehen Sie diese Option für große Verzeichnisse in Betracht, die nicht gescannt werden müssen.
6. (Optional) Wählen Sie für Docker-Anmeldeinformationen die Anmeldeinformationen aus, die Sie in Schritt 3 hinzugefügt haben.
7. Wählen Sie für die Region aus AWS-Region, an die die Scananforderung gesendet werden soll.
8. Konfigurieren Sie unter AWS-Authentifizierungsoptionen (wählen Sie eine) eine der folgenden Optionen. Wenn Sie alle leer lassen, verwendet das Plugin die Standard-Credential-Provider-Kette auf dem Jenkins Controller.

   - Geben Sie für IAM-Rolle den Amazon-Ressourcennamen (ARN) der Rolle ein, die Sie in Schritt 1 erstellt haben, z. B. `arn:aws:iam::111122223333:role/InspectorCICDscan-role` Um die Rolle bei OIDC zu übernehmen, wählen Sie auch die Anmeldeinformationen aus, die Sie in Schritt 4 für OIDC-Anmeldeinformationen hinzugefügt haben.
   - Wählen Sie für AWS-Anmeldeinformationen den Zugriffsschlüssel aus, den Sie in Schritt 4 hinzugefügt haben.
   - Geben Sie für AWS-Profilname den Namen eines Profils auf dem Jenkins Controller ein.

   Wenn Sie mehr als eine Option konfigurieren, verwendet das Plugin die erste Option in dieser Reihenfolge: AWS-Anmeldeinformationen, IAM-Rolle mit OIDC-Anmeldeinformationen, IAM-Rolle und dann AWS-Profilname.

9. (Optional) Wählen Sie unter Ausgabefilter scannen die Prüfungen aus, bei denen der Build fehlschlagen kann. Eine Beschreibung der einzelnen Optionen finden Sie unterAusgabefilter scannen.
10. Wählen Sie Speichern.

Amazon Inspector Scan zu einem hinzufügen Jenkins Pipeline

Die folgende deklarative Pipeline lädt den Amazon Inspector SBOM Generator automatisch herunter und authentifiziert sich mit einer IAM-Rolle. Der Build schlägt fehl, wenn beim Scan mehr Sicherheitslücken gefunden werden, als es die Schweregradschwellenwerte zulassen. *IMAGE_ID*Ersetzen Sie es durch Ihr Bild, wie`alpine:latest`, und *REGION* durch Ihr AWS-Region, wie`us-east-1`. *IAM_ROLE_ARN*Ersetzen Sie durch den ARN der Rolle, die Sie in Schritt 1 erstellt haben.

```
pipeline {
    agent any
    stages {
        stage('Amazon Inspector scan') {
            steps {
                script {
                    step([
                        $class:
 'com.amazon.inspector.jenkins.amazoninspectorbuildstep.AmazonInspectorBuilder',
                        sbomgenSelection: 'automatic',
                        archivePath: 'IMAGE_ID',
                        awsRegion: 'REGION',
                        iamRole: 'IAM_ROLE_ARN',
                        credentialId: '',
                        isSeverityThresholdEnabled: true,
                        countCritical: 0,
                        countHigh: 0,
                        countMedium: 5,
                        countLow: 10
                    ])
                }
            }
        }
    }
}
```

Wenn Sie eine `inspector-sbomgen` Binärdatei verwenden möchten, die Sie installiert haben, `sbomgenSelection: 'automatic'` ersetzen Sie sie durch die folgenden Zeilen. *`SBOMGEN_PATH`*Ersetzen Sie durch den vollständigen Pfad zur Binärdatei, z. `/opt/inspector/inspector-sbomgen` B.

```
sbomgenSelection: 'manual',
sbomgenPath: 'SBOMGEN_PATH',
```

Der Schritt akzeptiert die folgenden Parameter. Jeder Parameter entspricht einer Option des Build-Schritts.

Parameter der Amazon Inspector Scan-Pipeline

Parameter	Beschreibung
`archivePath`	Erforderlich Das zu scannende Bild (Bild-ID).
`awsRegion`	Erforderlich Das AWS-Region , an das die Scananforderung gesendet werden soll.
`sbomgenSelection`	`automatic` (Standard) oder`manual`. Jeder andere Wert verwendet`automatic` .
`sbomgenPath`	Der vollständige Pfad zur `inspector-sbomgen` Binärdatei. Erforderlich, wann `sbomgenSelection` ist`manual`.
`iamRole`	Der ARN der IAM-Rolle, die übernommen werden soll.
`oidcCredentialId`	Die ID der OIDC-Anmeldeinformationen, die das Plugin für die Annahme verwendet. `iamRole`
`awsCredentialId`	Die ID der Zugangsschlüssel-Anmeldeinformationen, die Sie in Schritt 4 hinzugefügt haben.
`awsProfileName`	Der Name eines Profils auf dem Jenkins Controller.

Parameter	Beschreibung
`credentialId`	Die ID der Docker Anmeldeinformationen für ein privates Repository. Verwenden Sie einen leeren Wert (`''`) für ein öffentliches Bild.
`reportArtifactName`	Ein Präfix für den Dateinamen der SBOM, die der Build archiviert. Der Standardwert ist `default-report` .
`sbomgenSkipFiles`	Die Dateien und Verzeichnisse, die vom Scan ausgeschlossen werden sollen, eine pro Zeile oder getrennt durch Kommas ohne Leerzeichen.
`isSeverityThresholdEnabled` , `countCritical` , `countHigh` , `countMedium` , `countLow`	Schwellenwerte für den Schweregrad. Eine Anzahl, die Sie nicht festlegen, ist`0`.
`isEpssThresholdEnabled` , `epssThreshold`	EPSS-Schwellenwert. Wenn Sie ihn ohne Einstellung aktivieren`epssThreshold` , überspringt das Plugin die Prüfung.
`isSuppressedCveEnabled` , `suppressedCveList`	Liste der CVE-Unterdrückung.
`isAutoFailCveEnabled` , `autoFailCveList`	Liste mit automatischen Ausfällen in CVE.
`isMaliciousPackageBlockingEnabled`	Blockierung bösartiger Pakete.
`isLicenseCollectionEnabled`	Sammlung von Lizenzen.

Eine Beschreibung des Schwellenwerts und der Filterparameter finden Sie unterAusgabefilter scannen.

> **Note**
>
> Das Plug-in akzeptiert weiterhin die früheren Parameternamen `isThresholdEnabled` und`isEpssEnabled`, protokolliert jedoch eine Warnung, die auf die aktuellen Namen hindeutet, `isSeverityThresholdEnabled` und`isEpssThresholdEnabled`.

Ausgabefilter scannen

Die folgenden Optionen steuern, wann der Scan beim Build fehlschlägt. Jede Option ist standardmäßig deaktiviert. Wenn Sie mehrere aktivieren, schlägt der Build fehl, falls einer von ihnen fehlschlägt. Die Konsolenausgabe des Builds zeigt, welche Optionen aktiviert sind und warum der Build fehlgeschlagen ist.

Aktivieren Sie Schweregradschwellenwerte () `isSeverityThresholdEnabled`

Der Build schlägt fehl, wenn die Anzahl der Ergebnisse eines Schweregrads den Schwellenwert überschreitet. Sie können einen Schwellenwert für Kritisch (`countCritical`), Hoch (`countHigh`), Mittel (`countMedium`) und Niedrig (`countLow`) festlegen. Die Anzahl umfasst Dockerfile-Ergebnisse, und jedes Ergebnis zählt einmal, und zwar bei seinem höchsten Schweregrad. Ein Schwellenwert von `0` schlägt beim Build fehl, wenn der Scan etwas mit diesem Schweregrad findet. Ergebnisse mit anderen Schweregraden werden nicht auf die Schwellenwerte angerechnet.

EPSS-Schwellenwert aktivieren (,) `isEpssThresholdEnabled` `epssThreshold`

Der Build schlägt fehl, wenn für eine Sicherheitslücke ein EPSS-Wert größer oder gleich dem Schwellenwert ist. Sicherheitslücken ohne EPSS-Score zählen nicht. Geben Sie einen Wert zwischen 0 und 1 ein, z. B. `0.7`

Aktivieren Sie die CVE-Unterdrückungsliste (`isSuppressedCveEnabled`,`suppressedCveList`)

Schließt die von Ihnen aufgelisteten CVEs von den Schweregrad- und EPSS-Schwellenwerten aus, z. B. falsch positive Ergebnisse, die Sie überprüft haben. Unterdrückte CVEs werden weiterhin in den Berichten angezeigt. Trennen Sie CVE-IDs durch Kommas oder Zeilenumbrüche, z. B. `CVE-2023-1234,CVE-2023-5678`

CVE-Autofail-Liste aktivieren (`isAutoFailCveEnabled`,) `autoFailCveList`

Der Build schlägt fehl, wenn der Scan ein von Ihnen aufgelistetes CVE findet, unabhängig von Ihren anderen Einstellungen, einschließlich der CVE-Unterdrückungsliste. Verwenden Sie diese

Liste für Sicherheitslücken, die niemals bereitgestellt werden dürfen. Trennen Sie CVE-IDs durch Kommas oder Zeilenumbrüche, z. B. `CVE-2024-9999`

Blockierung bösartiger Pakete aktivieren () `isMaliciousPackageBlockingEnabled`

Der Build schlägt fehl, wenn Amazon Inspector bösartige Pakete im Image identifiziert, unabhängig von Ihren anderen Einstellungen. Wenn das Scanergebnis keine schädlichen Paketinformationen enthält, überspringt das Plugin diese Prüfung mit einer Meldung in der Konsolenausgabe und schlägt den Build nicht fehl. Diese Option ist in der Plugin-Version 649.v0fe6e5596568 und höher verfügbar. Informationen darüber, wie Amazon Inspector bösartige Pakete identifiziert, finden Sie unterSicherheitsforschung von Amazon Inspector.

Lizenzsammlung aktivieren (`isLicenseCollectionEnabled`)

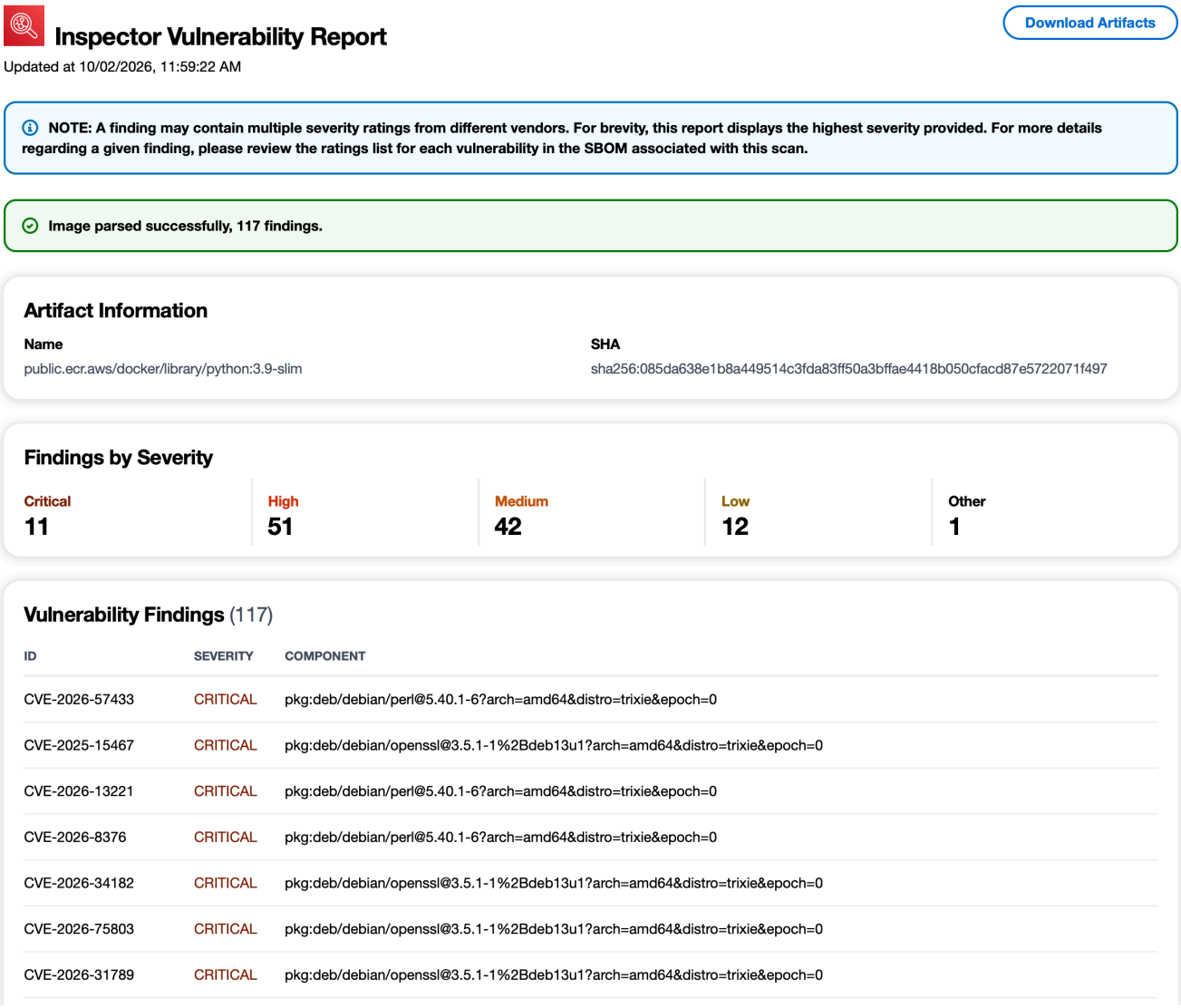
Fügt der SBOM, die der Build archiviert, Paketlizenzinformationen hinzu. Bei dieser Option schlägt der Build nicht fehl. Bei der manuellen Installationsmethode ist Amazon Inspector SBOM Generator Version 1.8.0 oder höher erforderlich. Weitere Informationen finden Sie unter Sammlung von Amazon Inspector SBOM Generator-Lizenzen.

Schritt 7. Sehen Sie sich Ihren Amazon Inspector Inspector-Schwachstellenbericht an

1. Führen Sie einen Build Ihres Projekts aus.
2. Öffnen Sie den Build. Wenn der Scan abgeschlossen ist, endet die Konsolenausgabe mit einer Zusammenfassung der Sicherheitsbeurteilung und gibt an, ob der Build bestanden hat.
3. Wählen Sie unter Build Artifacts aus`index.html`, ob Sie den HTML-Bericht öffnen möchten. Der Build archiviert auch die folgenden Dateien:

   - Die SBOM, die der Amazon Inspector SBOM Generator erstellt hat, im JSON-Format.
   - Eine CSV-Datei mit den Ergebnissen der Sicherheitslücke, falls beim Scan welche gefunden wurden.
   - Eine CSV-Datei mit den Dockerfile-Ergebnissen, falls beim Scan welche gefunden wurden.

   Der Bericht zeigt das gescannte Bild, die Anzahl der Ergebnisse für jeden Schweregrad und eine Tabelle, in der jede Sicherheitslücke mit ihrer ID, ihrem Schweregrad und dem betroffenen Paket aufgeführt ist. Die folgende Abbildung zeigt einen Beispielbericht.



Inspector Vulnerability Report

Download Artifacts

Updated at 10/02/2026, 11:59:22 AM

NOTE: A finding may contain multiple severity ratings from different vendors. For brevity, this report displays the highest severity provided. For more details regarding a given finding, please review the ratings list for each vulnerability in the SBOM associated with this scan.

Image parsed successfully, 117 findings.

Artifact Information

Name
public.ecr.aws/docker/library/python:3.9-slim

SHA
sha256:085da638e1b8a449514c3fda83ff50a3bffae4418b050cfacd87e5722071f497

Findings by Severity

Critical	High	Medium	Low	Other
11	51	42	12	1

Vulnerability Findings (117)

ID	SEVERITY	COMPONENT
CVE-2026-57433	CRITICAL	pkg:deb/debian/perl@5.40.1-6?arch=amd64&distro=trixie&epoch=0
CVE-2025-15467	CRITICAL	pkg:deb/debian/openssl@3.5.1-1%2Bdeb13u1?arch=amd64&distro=trixie&epoch=0
CVE-2026-13221	CRITICAL	pkg:deb/debian/perl@5.40.1-6?arch=amd64&distro=trixie&epoch=0
CVE-2026-8376	CRITICAL	pkg:deb/debian/perl@5.40.1-6?arch=amd64&distro=trixie&epoch=0
CVE-2026-34182	CRITICAL	pkg:deb/debian/openssl@3.5.1-1%2Bdeb13u1?arch=amd64&distro=trixie&epoch=0
CVE-2026-75803	CRITICAL	pkg:deb/debian/openssl@3.5.1-1%2Bdeb13u1?arch=amd64&distro=trixie&epoch=0
CVE-2026-31789	CRITICAL	pkg:deb/debian/openssl@3.5.1-1%2Bdeb13u1?arch=amd64&distro=trixie&epoch=0

Fehlerbehebung

Im Folgenden sind häufig auftretende Fehler aufgeführt, die bei der Verwendung des Amazon Jenkins Inspector-Plug-ins auftreten können. Wenn das Plugin fehlschlägt, schlägt der Build fehl und die Konsolenausgabe wird angezeigt, `Plugin execution failed.` gefolgt von dem Fehler.

Fehler beim Laden der Anmeldeinformationen

Fehler:

`software.amazon.awssdk.core.exception.SdkClientException: Unable to load credentials from any of the providers in the chain`, gefolgt von der Liste der Anmeldeinformationsanbieter, die das Plugin ausprobiert hat. Wenn Sie keine IAM-Rolle

festgelegt haben, kann stattdessen der letzte Fehler auftreten. `Profile file contained no credentials for profile 'default'`

Frühere Versionen des Plugins können diesen Fehler als anzeigen `InstanceProfileCredentialsProvider(): Failed to load credentials or sts exception.`

Auflösung

Das Plugin konnte keine AWS Anmeldeinformationen auf dem Jenkins Controller abrufen. Konfigurieren Sie eine der Authentifizierungsoptionen im Build-Schritt, wie in Schritt 4 und Schritt 6 beschrieben. Alternativ können Sie dem Controller Anmeldeinformationen zur Verfügung stellen, z. B. mit einem Amazon EC2 EC2-Instance-Profil. Wenn die Scananforderung fehlschlägt, versucht das Plugin erneut, ohne den Zugriffsschlüssel, das OIDC-Token oder den Profilnamen anzugeben. Wenn Sie eine IAM-Rolle festlegen, übernimmt sie die Rolle mit den eigenen Anmeldeinformationen des Controllers. Andernfalls wird das `default` Profil auf dem Controller verwendet. Aufgrund dieses Wiederholungsversuchs zeigt der erste Fehler in der Konsolenausgabe in der Regel die Ursache an.

Das Bild konnte nicht aus Tarball-, lokalen oder Remote-Quellen geladen werden

Fehler:

`[ImageDownloadFailed]: failed to load image from tarball, local, or remote sources.`

Je nach Ursache kann der Amazon Inspector SBOM Generator stattdessen einen spezifischeren Fehler melden, z. B. `[ImageDownloadManifestUnknown]` oder. `[ImageRegistryAccessUnauthorized]` In der Konsolenausgabe wird der Amazon Inspector SBOM Generator-Fehler angezeigt`Exception:com.amazon.inspector.jenkins.amazoninspectorbuildstep.exception Sbom scanning output formatted incorrectly.`, gefolgt von `Sbom Content:` und.

Auflösung

Überprüfen Sie Folgendes:

- Der Benutzer, der den Build ausführt, hat Leseberechtigungen für das Bild, das Sie scannen möchten.
- Bei einem lokalen Image ist das Image in der Docker Engine auf dem Computer vorhanden, auf dem der Build ausgeführt wird.

- Der Image-Name ist korrekt.
- Für ein Image in einem privaten Repository verwendet der Build-Schritt die Docker Anmeldeinformationen aus Schritt 3.

Inspector-sbomgen Pfadfehler

Fehler:

```
Exception:java.lang.IllegalArgumentException: Provided SBOMgen path is
invalid or not executable: /opt/inspector/inspector-sbomgen
```

Frühere Versionen des Plugins zeigen stattdessen den folgenden Fehler:
`Exception:com.amazon.inspector.jenkins.amazoninspectorbuildstep.exception.Sbomge`
`There was an issue running inspector-sbomgen, is /opt/inspector/inspector-sbomgen the correct path?`

Auflösung

Gehen Sie wie folgt vor, um das Problem zu beheben.

1. Laden Sie den Amazon Inspector SBOM Generator für die CPU-Architektur des Computers herunter, auf dem der Build ausgeführt wird. Weitere Informationen finden Sie unter Installation Sbomgen.
2. Machen Sie die Binärdatei ausführbar, indem Sie den folgenden Befehl ausführen:. `chmod +x inspector-sbomgen`
3. Geben Sie im Build-Schritt den vollständigen Pfad zur Binärdatei ein, z. `/opt/inspector/inspector-sbomgen` B. Das Plugin überprüft den Pfad auf dem Jenkins Controller. Wenn der Build also auf einem Agenten ausgeführt wird, muss sich die Binärdatei auf beiden im gleichen Pfad befinden.
4. Speichern Sie die Konfiguration und führen Sie den Build erneut aus.

Wenn der Fehler auftritt`Invalid sbomgen path: path`, enthält der Pfad ein Zeichen, das nicht zulässig ist. Der Pfad darf nur Buchstaben, Zahlen, Leerzeichen, Punkte (.), Unterstriche (_), Bindestriche (-), Doppelpunkte (:) und Schrägstriche (/) enthalten.

Fehler: Betriebssystem oder Architektur nicht unterstützt

Fehler:

```
Exception:java.lang.UnsupportedOperationException: Unsupported OS: mac os x
```

Der Fehler kann auch sein. `Exception:java.lang.UnsupportedOperationException: Unsupported architecture: architecture`

Auflösung

Der Amazon Inspector SBOM Generator ist nur für Linux eine x86_64- (amd64) - oder arm64-CPU verfügbar. Bei der automatischen Installationsmethode überprüft das Plugin das Betriebssystem des Jenkins Controllers und die CPU des Computers, auf dem der Build ausgeführt wird. Führen Sie den Controller auf Linux und führen Sie Ihren Build auf Linux einer unterstützten CPU aus.

Der HTML-Bericht hat kein Design oder zeigt keine Ergebnisse

Fehler:

Der HTML-Bericht hat kein Design und zeigt keine Ergebnisse. Der obere Teil des Berichts zeigt `Jenkins may be blocking CSS/JS, visit https://www.jenkins.io/doc/book/security/configuring-content-security-policy/ for a possible fix.`

Auflösung

Jenkinsblockiert das CSS und JavaScript den Bericht. Informationen zur Anzeige des Berichts finden Sie unterSchritt 5. Sobald Sie die Details auf dieser Seite überprüft haben, klicken Sie auf Jenkins um den HTML-Bericht anzuzeigen. In den CSV-Dateien werden die Ergebnisse auch dann angezeigt, wenn der HTML-Bericht nicht angezeigt wird.

Verwenden CodeCatalyst Aktionen mit Amazon Inspector

Sie können Amazon Inspector mit Amazon verwenden CodeCatalyst, um Amazon Inspector Inspector-Schwachstellenscans zu Ihren CodeCatalyst Workflows hinzuzufügen. Dabei werden der Amazon Inspector SBOM Generator und die Amazon Inspector Scan API genutzt, um am Ende Ihres Builds detaillierte Berichte zu erstellen, sodass Sie Risiken vor der Bereitstellung untersuchen und beheben können. Amazon Inspector Vulnerability Scans können je nach Anzahl und Schweregrad der erkannten Sicherheitslücken so konfiguriert werden, dass Workflows erfolgreich oder nicht bestanden werden. Informationen zur Integration von Amazon Inspector Scan in Ihre CI/CD Pipeline finden Sie unter Amazon Inspector-Scans in Ihre CI/CD Pipeline integrieren. Eine Liste der Betriebssysteme und Programmiersprachen, die Amazon Inspector unterstützt, finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen.

Verwenden von Amazon Inspector Scan-Aktionen mit CodePipeline

Sie können Amazon Inspector mit verwenden, AWS CodePipeline indem Sie Ihren Workflows Schwachstellenscans hinzufügen. Diese Integration nutzt den Amazon Inspector SBOM Generator und die Amazon Inspector Scan API, um am Ende Ihres Builds detaillierte Berichte zu erstellen. Die Integration hilft Ihnen, Risiken vor der Bereitstellung zu untersuchen und zu beheben. Bei der `InspectorScan` Aktion handelt es sich um eine verwaltete Rechenaktion CodePipeline , mit der die Erkennung und Behebung von Sicherheitslücken in Ihrem Open-Source-Code automatisiert wird. Du kannst diese Aktion mit Anwendungsquellcode in deinem Drittanbieter-Repository wie Bitbucket Cloud GitHub oder mit Bildern für Containeranwendungen verwenden. Weitere Informationen findest du in der Referenz „Aktion InspectorScan aufrufen“ im AWS CodePipeline Benutzerhandbuch.

Den Amazon Inspector verwenden TeamCity Plugin

> **Note**
>
> Das Amazon TeamCity Inspector-Plugin ist eingestellt und archiviert amGitHub.

Das Amazon TeamCity Inspector-Plugin nutzt die Amazon Inspector SBOM Generator-Binärdatei und die Amazon Inspector Scan API, um am Ende Ihres Builds detaillierte Berichte zu erstellen, sodass Sie Risiken vor der Bereitstellung untersuchen und beheben können. Mit dem Amazon TeamCity Inspector-Plugin können Sie Amazon Inspector Inspector-Schwachstellenscans zu Ihrer TeamCity Pipeline hinzufügen. Amazon Inspector Vulnerability Scans können so konfiguriert werden, dass Pipeline-Ausführungen je nach Anzahl und Schweregrad der erkannten Sicherheitslücken bestanden oder fehlschlagen. Sie können die archivierte Version des Amazon TeamCity Inspector-Plug-ins GitHub unter einsehen https://github.com/aws/amazon-inspector-container-image-scanner-teamcity-plugin.

Verwenden von Amazon Inspector mit GitLab components

> **Note**
>
> Die Amazon Inspector GitLab Inspector-Komponente wird von Amazon nicht offiziell unterstützt. Sie wird von gepflegt und als Open Source bereitgestelltGitLab.

Sie können Amazon Inspector mit GitLab CI/CD Komponenten verwenden, um Amazon Inspector Inspector-Schwachstellenscans zu Ihren GitLab Projekten hinzuzufügen. Dabei werden der Amazon Inspector SBOM Generator und die Amazon Inspector Scan API genutzt, um am Ende Ihres Builds detaillierte Berichte zu erstellen, sodass Sie Risiken vor der Bereitstellung untersuchen und beheben können. Amazon Inspector Vulnerability Scans können je nach Anzahl und Schweregrad der erkannten Sicherheitslücken so konfiguriert werden, dass Workflows erfolgreich oder nicht bestanden werden. Sie können die neueste Version der Amazon Inspector Inspector-Komponente auf der GitLabWebsite einsehen. Informationen zur Integration von Amazon Inspector Scan in Ihre CI/CD Pipeline finden Sie unter Amazon Inspector-Scans in Ihre CI/CD Pipeline integrieren. Eine Liste der Betriebssysteme und Programmiersprachen, die Amazon Inspector unterstützt, finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen.

Bewertung der Abdeckung Ihrer AWS Umgebung durch Amazon Inspector

Sie können den Umfang Ihrer AWS Umgebung durch Amazon Inspector auf dem Bildschirm Kontoverwaltung in der Amazon Inspector-Konsole beurteilen. Dort werden Details und Statistiken zum Status der Amazon Inspector-Scans für Ihre Konten und Ressourcen angezeigt.

> **Note**
>
> Wenn Sie der delegierte Administrator einer Organisation sind, können Sie Details und Statistiken für alle Konten in der Organisation einsehen.

Das folgende Verfahren beschreibt, wie Sie die Abdeckung Ihrer Amazon Inspector Inspector-Umgebung beurteilen können.

Um die Abdeckung Ihrer AWS Umgebung durch Amazon Inspector zu beurteilen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich Kontoverwaltung aus.
3. Um den Versicherungsschutz zu überprüfen, wählen Sie eine der folgenden Registerkarten:

   - Wählen Sie Konten, um den Versicherungsschutz auf Kontoebene zu überprüfen.
   - Wählen Sie Instances aus, um den Versicherungsschutz für Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu überprüfen.
   - Wählen Sie Container-Repositorys aus, um den Umfang der Amazon Elastic Container Registry (Amazon ECR) -Repositorys zu überprüfen.
   - Wählen Sie Container-Images aus, um die Abdeckung von Amazon ECR-Container-Images zu überprüfen.
   - Wählen Sie Lambda-Funktionen, um den Umfang der Lambda-Funktionen zu überprüfen.

In den folgenden Themen werden die Informationen beschrieben, die jede dieser Registerkarten enthält.

Themen

- Bewertung der Deckung auf Kontoebene
- Bewertung der Abdeckung von Amazon EC2 EC2-Instances
- Bewertung der Abdeckung von Amazon ECR-Repositorien
- Bewertung der Reichweite von Amazon ECR-Container-Images
- Bewertung des Funktionsumfangs AWS Lambda

Bewertung der Deckung auf Kontoebene

Wenn Ihr Konto nicht Teil einer Organisation ist oder nicht das delegierte Amazon Inspector-Administratorkonto für eine Organisation ist, finden Sie auf der Registerkarte Konten Informationen über Ihr Konto und den Status des Ressourcenscans für Ihr Konto. Auf dieser Registerkarte können Sie das Scannen für alle oder nur bestimmte Arten von Ressourcen für Ihr Konto aktivieren oder deaktivieren. Weitere Informationen finden Sie unter Automatisierte Scantypen in Amazon Inspector.

Wenn Ihr Konto das delegierte Amazon Inspector-Administratorkonto für eine Organisation ist, bietet die Registerkarte Konten automatische Aktivierungseinstellungen für Konten in Ihrer Organisation und listet alle Konten in Ihrer Organisation auf. Für jedes Konto gibt die Liste an, ob Amazon Inspector für das Konto aktiviert ist, und wenn ja, welche Arten von Ressourcenscans für das Konto aktiviert sind. Als delegierter Administrator können Sie auf dieser Registerkarte die Einstellungen für die automatische Aktivierung für Ihr Unternehmen ändern. Sie können auch bestimmte Arten des Ressourcenscans für einzelne Mitgliedskonten aktivieren oder deaktivieren. Weitere Informationen finden Sie unter Die Aktivierung von Amazon Inspector scannt nach Mitgliedskonten.

Bewertung der Abdeckung von Amazon EC2 EC2-Instances

Auf der Registerkarte Instances werden Amazon EC2 EC2-Instances in Ihrer AWS Umgebung angezeigt. Die Listen sind auf den folgenden Registerkarten in Gruppen unterteilt:

- Alle — Zeigt alle Instanzen in Ihrer Umgebung an. In der Spalte Status wird der aktuelle Scanstatus für eine Instance angezeigt.
- Scannen — Zeigt alle Instances an, die Amazon Inspector in Ihrer Umgebung aktiv überwacht und scannt.
- Nicht scannen — Zeigt alle Instances an, die Amazon Inspector in Ihrer Umgebung nicht überwacht und scannt. Die Spalte Grund gibt an, warum Amazon Inspector eine Instance nicht überwacht und scannt.

Eine EC2-Instance kann aus verschiedenen Gründen auf der Registerkarte Nicht gescannt angezeigt werden. Amazon Inspector verwendet AWS Systems Manager (SSM) und den SSM-Agenten, um Ihre EC2-Instances automatisch zu überwachen und auf Sicherheitslücken zu scannen. Wenn auf einer Instance der SSM-Agent nicht ausgeführt wird, sie keine AWS Identity and Access Management (IAM-) Rolle hat, die Systems Manager unterstützt, oder auf der kein unterstütztes Betriebssystem oder keine unterstützte Architektur ausgeführt wird, kann Amazon Inspector die Instance nicht überwachen und scannen. Weitere Informationen finden Sie unter Scannen Amazon EC2 EC2-Instances.

Auf jeder Registerkarte gibt die Spalte Konto an, wer Eigentümer einer AWS-Konto Instance ist.

EC2-Instance-Tags — In dieser Spalte werden die mit der Instance verknüpften Tags angezeigt. Anhand dieser Spalte können Sie feststellen, ob Ihre Instance anhand von Stichwörtern von Scans ausgeschlossen wurde.

Betriebssystem — In dieser Spalte wird der Betriebssystemtyp angezeigt. Dieser kann, `WINDOWS` `MACLINUX`, oder `UNKNOWN` sein.

Überwacht mit — In dieser Spalte wird angezeigt, ob Amazon Inspector für diese Instance die agentenbasierte oder die agentenlose Scanmethode verwendet.

Zuletzt gescannt — In dieser Spalte wird angezeigt, wann Amazon Inspector diese Ressource zuletzt auf Sicherheitslücken überprüft hat. Die Häufigkeit, mit der Amazon Inspector Scans durchführt, hängt von der Scanmethode ab, die zum Scannen der Instance verwendet wird.

Um weitere Details zu einer EC2-Instance zu überprüfen, wählen Sie den Link in der Spalte EC2-Instance. Amazon Inspector zeigt dann Details zur Instance und aktuelle Ergebnisse für die Instance an. Um die Details eines Ergebnisses zu überprüfen, wählen Sie den Link in der Titelspalte. Informationen zu diesen Details finden Sie unterDetails zu Ihren Amazon Inspector Inspector-Ergebnissen anzeigen.

Statuswerte für Amazon EC2 EC2-Instances scannen

Für eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance sind die möglichen Statuswerte wie folgt:

- Aktive Überwachung — Amazon Inspector überwacht und scannt die Instance kontinuierlich.

- Speicherlimit für agentenlose Instances überschritten — Amazon Inspector verwendet diesen Status, wenn die Gesamtgröße aller an eine Instance angehängten Volumes mehr als 1200 GB beträgt oder wenn an eine Instance mehr als 8 Volumes angehängt sind.
- Zeitlimit für die Erfassung agentenloser Instances überschritten — Amazon Inspector tritt beim Versuch, einen agentenlosen Scan auf einer Instance auszuführen, ein Timeout auf.
- EC2-Instance gestoppt — Amazon Inspector hat die Suche nach der Instance angehalten, da sich die Instance in einem gestoppten Zustand befindet. Alle vorhandenen Ergebnisse bleiben bestehen, bis die Instance beendet wird. Wenn die Instance neu gestartet wird, fährt Amazon Inspector automatisch mit dem Scannen nach der Instance fort.
- Interner Fehler — Ein interner Fehler ist aufgetreten, als Amazon Inspector versuchte, die Instance zu scannen. Amazon Inspector behebt den Fehler automatisch und setzt den Scanvorgang so schnell wie möglich fort.
- Kein Inventar — Amazon Inspector konnte das Inventar der Softwareanwendung, das nach der Instance gescannt werden soll, nicht finden. Die Amazon Inspector Inspector-Verknüpfungen für die Instance wurden möglicherweise gelöscht oder konnten möglicherweise nicht ausgeführt werden.

  Um dieses Problem zu beheben, stellen Sie bitte AWS Systems Manager sicher, dass die `InspectorInventoryCollection-do-not-delete` Zuordnung besteht und ihr Zuordnungsstatus erfolgreich ist. Verwenden Sie außerdem AWS Systems Manager Fleet Manager, um das Softwareanwendungsinventar für die Instanz zu überprüfen.
- Ausstehende Deaktivierung — Amazon Inspector hat das Scannen der Instance beendet. Die Instance wird deaktiviert, bis die Bereinigungsaufgaben abgeschlossen sind.
- Erster Scan steht aus — Amazon Inspector hat die Instance für einen ersten Scan in die Warteschlange gestellt.
- Ressource beendet — Die Instance wurde beendet. Amazon Inspector bereinigt derzeit die vorhandenen Ergebnisse und Deckungsdaten für die Instance.
- Veraltetes Inventar — Amazon Inspector war nicht in der Lage, ein aktualisiertes Softwareanwendungsinventar zu sammeln, das innerhalb der letzten 7 Tage für die Instance erfasst wurde.

  Um dieses Problem zu beheben, stellen Sie AWS Systems Manager sicher, dass die erforderlichen Amazon Inspector Inspector-Verknüpfungen vorhanden sind und für die Instance ausgeführt werden. Verwenden Sie außerdem AWS Systems Manager Fleet Manager, um den Bestand der Softwareanwendungen für die Instance zu überprüfen.

- Nicht verwaltete EC2-Instance — Amazon Inspector überwacht oder scannt die Instance nicht. Die Instance wird nicht von verwaltet. AWS Systems Manager

  Um dieses Problem zu beheben, können Sie das von AWS Systems Manager Automation AWSSupport-TroubleshootManagedInstance runbookbereitgestellte Tool verwenden. Nachdem Sie die Konfiguration AWS Systems Manager für die Verwaltung der Instance vorgenommen haben, beginnt Amazon Inspector automatisch, die Instance kontinuierlich zu überwachen und zu scannen.
- Nicht unterstütztes Betriebssystem — Amazon Inspector überwacht oder scannt die Instance nicht. Die Instance verwendet ein Betriebssystem oder eine Architektur, die Amazon Inspector nicht unterstützt. Eine Liste der Betriebssysteme, die Amazon Inspector unterstützt, finden Sie unterStatuswerte Amazon EC2 EC2-Instances.
- Aktive Überwachung mit teilweisen Fehlern — Dieser Status bedeutet, dass das EC2-Scannen aktiv ist, aber es liegen Fehler vor. Eingehende Inspektion von Amazon Inspector für Linux-based Amazon EC2 EC2-Instances Bei tiefgreifenden Inspektionen kann es sich um folgende Fehler handeln:
  - Das Limit für die Erfassung von Paketen mit Tiefeninspektion wurde überschritten — Die Instance hat das Limit von 5000 Paketen für die Tiefeninspektion von Amazon Inspector überschritten. Um die Tiefeninspektion für diese Instance fortzusetzen, können Sie versuchen, die mit dem Konto verknüpften benutzerdefinierten Pfade anzupassen.
  - Das tägliche SSM-Inventarlimit für Deep Inspection wurde überschritten — Der SSM-Agent konnte kein Inventar an Amazon Inspector senden, da das SSM-Kontingent für die pro Instance und Tag gesammelten SSM-Inventardaten für diese Instance bereits erreicht wurde. Weitere Informationen finden Sie unter Amazon EC2 Systems Manager Manager-Endpunkte und Kontingente.
  - Zeitlimit für die Abholung bei Tiefeninspektion überschritten — Amazon Inspector konnte den Paketbestand nicht extrahieren, da die Paketabholzeit den maximalen Schwellenwert von 15 Minuten überschritten hat.
  - Deep Inspection hat kein Inventar — Das Amazon Inspector SSM-Plugin war noch nicht in der Lage, ein Inventar von Paketen für diese Instanz zu sammeln. Dies ist normalerweise das Ergebnis eines ausstehenden Scans. Wenn dieser Status jedoch nach 6 Stunden weiterhin besteht, verwenden Sie Amazon EC2 Systems Manager, um sicherzustellen, dass die erforderlichen Amazon Inspector Inspector-Verknüpfungen vorhanden sind und für die Instance ausgeführt werden. Dieser Fehler kann auch auftreten, wenn der Instance die erforderlichen `ssm:PutInventory` und `ssm:GetParameter` erforderlichen Berechtigungen fehlen. Wenn die Instanz über ein IAM-Instanzprofil verfügt, überprüfen Sie, ob diese Berechtigungen im

Profil enthalten sind. Wenn kein Instanzprofil konfiguriert ist, stellen Sie sicher, dass diese Berechtigungen in der Rolle „Standard-Host-Management-Konfiguration“ enthalten sind.

Einzelheiten zur Konfiguration der Scaneinstellungen für eine EC2-Instance finden Sie unterScannen Amazon EC2 EC2-Instances.

Bewertung der Abdeckung von Amazon ECR-Repositorien

Auf der Registerkarte Repositorys werden Amazon ECR-Repositorys in Ihrer Umgebung angezeigt. AWS Die Listen sind auf den folgenden Registerkarten in Gruppen unterteilt:

- Alle — Zeigt alle Repositorys in Ihrer Umgebung an. In der Spalte Status wird der aktuelle Scanstatus für ein Repository angezeigt.
- Aktiviert — Zeigt alle Repositorys an, für deren Überwachung und Scannen Amazon Inspector in Ihrer Umgebung konfiguriert ist. Die Spalte Status zeigt den aktuellen Scanstatus für ein Repository an.
- Nicht aktiviert — Zeigt alle Repositorys an, die Amazon Inspector in Ihrer Umgebung nicht überwacht und scannt. Die Spalte Grund gibt an, warum Amazon Inspector ein Repository nicht überwacht und scannt.

Auf jeder Registerkarte gibt die Spalte Konto an, wer Eigentümer eines Repositorys ist. AWS-Konto

Um weitere Details zu einem Repository zu überprüfen, wählen Sie den Namen des Repositorys. Amazon Inspector zeigt dann eine Liste der Container-Images im Repository sowie Details zu jedem Bild an. Zu den Details gehören das Image-Tag, der Image-Digest und der Scanstatus. Sie enthalten auch wichtige Ergebnisstatistiken, wie z. B. die Anzahl kritischer Ergebnisse für das Bild. Wählen Sie das Bild-Tag für das Bild aus, um weitere Informationen zu erhalten und die unterstützenden Daten für die Suche nach Statistiken zu überprüfen.

> **Note**
>
> Amazon ECR-Bilder ohne kontinuierliches Scannen sind nicht in den Coverage-Widgets enthalten.

Statuswerte für Amazon ECR-Repositorys scannen

Für ein Amazon Elastic Container Registry (Amazon ECR) -Repository sind die möglichen Statuswerte:

- Aktiviert (Kontinuierlich) — Für ein Repository überwacht Amazon Inspector kontinuierlich die Bilder in diesem Repository. Die erweiterte Scan-Einstellung für das Repository ist auf kontinuierliches Scannen eingestellt. Amazon Inspector scannt neue Bilder zunächst, wenn sie übertragen werden, und scannt Bilder erneut, wenn ein neues CVE veröffentlicht wird, das für dieses Bild relevant ist. Amazon Inspector überwacht weiterhin die Bilder in diesem Repository für die Dauer des Amazon ECR-Neuscans, die Sie konfigurieren.
- Aktiviert (bei Push) — Amazon Inspector scannt automatisch einzelne Container-Images im Repository, wenn ein neues Image übertragen wird. Das erweiterte Scannen ist für das Repository aktiviert und so eingestellt, dass bei Push gescannt wird.
- Zugriff verweigert — Amazon Inspector darf weder auf das Repository noch auf Container-Images im Repository zugreifen.

  Um dieses Problem zu beheben, stellen Sie sicher, dass AWS Identity and Access Management (IAM-) Richtlinien für das Repository Amazon Inspector den Zugriff auf das Repository ermöglichen.
- Deaktiviert (Manuell) — Amazon Inspector überwacht oder scannt keine Container-Images im Repository. Die Amazon ECR-Scaneinstellung für das Repository ist auf einfaches manuelles Scannen eingestellt.

  Um mit dem Scannen von Bildern im Repository mit Amazon Inspector zu beginnen, ändern Sie die Scan-Einstellung für das Repository auf Erweitertes Scannen und wählen Sie dann, ob Bilder kontinuierlich oder nur dann gescannt werden sollen, wenn ein neues Bild übertragen wird.
- Aktiviert (bei Push) — Amazon Inspector scannt automatisch einzelne Container-Images im Repository, wenn ein neues Image übertragen wird. Die erweiterte Scan-Einstellung für das Repository ist auf Scan on Push eingestellt.
- Interner Fehler — Ein interner Fehler ist aufgetreten, als Amazon Inspector versuchte, das Repository zu scannen. Amazon Inspector behebt den Fehler automatisch und setzt den Scanvorgang so schnell wie möglich fort.

Für Einzelheiten zur Konfiguration der Scaneinstellungen für RepositorysScannen von Amazon ECR-Container-Bildern.

Bewertung der Reichweite von Amazon ECR-Container-Images

Auf der Registerkarte Images werden Amazon ECR-Container-Images in Ihrer AWS Umgebung angezeigt. Die Listen sind auf den folgenden Registerkarten in Gruppen unterteilt:

- Alle — Zeigt alle Container-Images in Ihrer Umgebung an. In der Spalte Status wird der aktuelle Scanstatus für ein Bild angezeigt.
- Scannen — Zeigt alle Container-Images an, für deren Überwachung und Scannen Amazon Inspector in Ihrer Umgebung konfiguriert ist. Die Spalte Status zeigt den aktuellen Scanstatus für ein Bild an.
- Nicht scannen — Zeigt alle Container-Images an, die Amazon Inspector in Ihrer Umgebung nicht überwacht und scannt. Die Spalte Grund gibt an, warum Amazon Inspector ein Bild nicht überwacht und scannt.

  Ein Container-Bild kann aus verschiedenen Gründen auf der Registerkarte Nicht aktiviert angezeigt werden. Das Bild ist möglicherweise in einem Repository gespeichert, für das Amazon Inspector-Scans nicht aktiviert sind, oder Amazon ECR-Filterregeln verhindern, dass dieses Repository gescannt wird. Oder das Bild wurde nicht innerhalb der Anzahl von Tagen übertragen oder abgerufen, die Sie für die Dauer des erneuten ECR-Scans konfiguriert haben. Weitere Informationen finden Sie unter Konfiguration der Dauer des Amazon ECR-Neuscans.

Auf jeder Registerkarte gibt die Spalte Repository-Name den Namen des Repositorys an, das ein Container-Image speichert. In der Spalte Konto wird der AWS-Konto Eigentümer des Repositorys angegeben. In der Spalte Zuletzt gescannt wird angezeigt, wann Amazon Inspector diese Ressource zuletzt auf Sicherheitslücken überprüft hat. Dies kann Prüfungen beinhalten, wenn die Suchmetadaten aktualisiert wurden, wenn das Anwendungsinventar der Ressource aktualisiert wurde oder wenn als Reaktion auf einen neuen CVE ein erneuter Scan durchgeführt wird. Weitere Informationen finden Sie unter Scanverhalten für Amazon ECR-Scans.

Um weitere Details zu einem Container-Image zu überprüfen, wählen Sie den Link in der Spalte ECR-Container-Image. Amazon Inspector zeigt dann Details zum Bild und aktuelle Ergebnisse für das Bild an. Um die Details eines Ergebnisses zu überprüfen, wählen Sie den Link in der Titelspalte. Informationen zu diesen Details finden Sie unterDetails zu Ihren Amazon Inspector Inspector-Ergebnissen anzeigen.

Statuswerte für Amazon ECR-Container-Images scannen

Für ein Amazon Elastic Container Registry-Container-Image sind die möglichen Statuswerte wie folgt:

- Aktive Überwachung (kontinuierlich) — Amazon Inspector überwacht kontinuierlich das Bild und es werden neue Scans durchgeführt, wenn ein neuer relevanter CVE veröffentlicht wird. Die Dauer des Amazon ECR-Rescans für das Bild wird jedes Mal aktualisiert, wenn das Bild übertragen oder abgerufen wird. Das erweiterte Scannen ist für das Repository aktiviert, in dem das Bild gespeichert ist, und die erweiterte Scan-Einstellung für das Repository ist auf kontinuierliches Scannen eingestellt.
- Aktiviert (bei Push) — Amazon Inspector scannt das Bild automatisch jedes Mal, wenn ein neues Bild übertragen wird. Das erweiterte Scannen ist für das Repository aktiviert, in dem das Bild gespeichert ist, und die erweiterte Scan-Einstellung für das Repository ist auf Scannen bei Push eingestellt.
- Interner Fehler — Ein interner Fehler ist aufgetreten, als Amazon Inspector versuchte, das Container-Image zu scannen. Amazon Inspector behebt den Fehler automatisch und setzt den Scanvorgang so schnell wie möglich fort.
- Erster Scan steht aus — Amazon Inspector hat das Bild für einen ersten Scan in die Warteschlange gestellt.
- Die Scanberechtigung ist abgelaufen (Fortlaufend) — Amazon Inspector hat den Scanvorgang für das Bild ausgesetzt. Das Bild wurde nicht innerhalb der Dauer aktualisiert, die Sie für automatische erneute Scans von Bildern im Repository angegeben haben. Sie können das Bild verschieben oder ziehen, um den Scanvorgang fortzusetzen.
- Die Scanberechtigung ist abgelaufen (On Push) — Amazon Inspector hat den Scanvorgang für das Bild ausgesetzt. Das Bild wurde nicht innerhalb des Zeitraums aktualisiert, den Sie für automatische erneute Scans von Bildern im Repository angegeben haben. Sie können das Bild per Push übertragen, um den Scanvorgang fortzusetzen.
- Manuelles Scannen der Frequenz (Manuell) — Amazon Inspector scannt das Amazon ECR-Container-Image nicht. Die Amazon ECR-Scaneinstellung für das Repository, in dem das Bild gespeichert ist, ist auf einfaches manuelles Scannen eingestellt. Um das automatische Scannen des Bilds mit Amazon Inspector zu starten, ändern Sie die Repository-Einstellung auf Verbessertes Scannen und wählen Sie dann, ob Bilder kontinuierlich oder nur gescannt werden sollen, wenn ein neues Bild übertragen wird.
- Nicht unterstütztes Betriebssystem — Amazon Inspector überwacht oder scannt das Bild nicht. Das Bild basiert auf einem Betriebssystem, das Amazon Inspector nicht unterstützt, oder es verwendet einen Medientyp, den Amazon Inspector nicht unterstützt.

Eine Liste der Betriebssysteme, die Amazon Inspector unterstützt, finden Sie unterUnterstützte Betriebssysteme: Amazon ECR-Scannen mit Amazon Inspector. Eine Liste der Medientypen, die Amazon Inspector unterstützt, finden Sie unter Unterstützte Medientypen.

Einzelheiten zur Konfiguration der Scaneinstellungen für Repositorys und Bilder finden Sie unterScannen von Amazon ECR-Container-Bildern.

Bewertung des Funktionsumfangs AWS Lambda

Auf der Registerkarte Lambda werden Lambda-Funktionen in Ihrer AWS Umgebung angezeigt. Auf dieser Seite gibt es zwei Tabellen, eine mit Details zur Funktionsabdeckung für das Lambda-Standardscannen und eine andere für das Scannen von Lambda-Code. Sie können Funktionen auf der Grundlage der folgenden Registerkarten gruppieren:

- Alle — Zeigt alle Lambda-Funktionen in Ihrer Umgebung an. Die Spalte Status zeigt den aktuellen Scanstatus für eine Lambda-Funktion an.
- Scannen — Zeigt die Lambda-Funktionen an, für deren Scannen Amazon Inspector konfiguriert ist. Die Spalte Status zeigt den aktuellen Scanstatus für jede Lambda-Funktion an.
- Nicht scannen — Zeigt die Lambda-Funktionen an, für deren Scannen Amazon Inspector nicht konfiguriert ist. Die Spalte Grund gibt an, warum Amazon Inspector eine Funktion nicht überwacht und scannt.

  Eine Lambda-Funktion kann aus verschiedenen Gründen auf der Registerkarte Nicht scannend angezeigt werden. Die Lambda-Funktion gehört möglicherweise zu einem Konto, das nicht zu Amazon Inspector hinzugefügt wurde, oder Filterregeln verhindern, dass diese Funktion gescannt wird. Weitere Informationen finden Sie unter Scannen mit Lambda-Funktionen.

Auf jeder Registerkarte gibt die Spalte Funktionsname den Namen der Lambda-Funktion an. In der Spalte Konto wird der Eigentümer AWS-Konto der Funktion angegeben. Runtime gibt die Laufzeit der Funktion an. Die Spalte Status zeigt den aktuellen Scanstatus für jede Lambda-Funktion an. Resource Tags zeigt die Tags an, die auf die Funktion angewendet wurden. In der Spalte Zuletzt gescannt wird angezeigt, wann Amazon Inspector diese Ressource zuletzt auf Sicherheitslücken überprüft hat. Dies kann Prüfungen beinhalten, wenn die Suchmetadaten aktualisiert wurden, wenn das Anwendungsinventar der Ressource aktualisiert wurde oder wenn als Reaktion auf einen neuen CVE ein erneuter Scan durchgeführt wird. Weitere Informationen finden Sie unter Scanverhalten beim Scannen mit Lambda-Funktionen.

Statuswerte für Funktionen werden gescannt AWS Lambda

Für eine Lambda-Funktion sind folgende Statuswerte möglich:

- Aktive Überwachung — Amazon Inspector überwacht und scannt kontinuierlich Lambda-Funktionen. Kontinuierliches Scannen umfasst einen ersten Scan neuer Funktionen, wenn sie in das Repository übertragen werden, und automatische erneute Scans von Funktionen, wenn sie aktualisiert werden oder wenn neue Common Vulnerabilities and Exposures (CVEs) veröffentlicht werden.
- Nach Tag ausgeschlossen — Amazon Inspector scannt diese Funktion nicht, da sie von Tag-Scans ausgeschlossen wurde.
- Die Scanberechtigung ist abgelaufen — Amazon Inspector überwacht diese Funktion nicht, da seit dem letzten Aufruf oder der letzten Aktualisierung mindestens 90 Tage vergangen sind.
- Interner Fehler — Beim Versuch von Amazon Inspector, die Funktion zu scannen, ist ein interner Fehler aufgetreten. Amazon Inspector behebt den Fehler automatisch und setzt den Scanvorgang so schnell wie möglich fort.
- Erster Scan steht aus — Amazon Inspector hat die Funktion für einen ersten Scan in die Warteschlange gestellt.
- Nicht unterstützt — Die Lambda-Funktion hat eine Laufzeit, die nicht unterstützt wird.

Verwaltung mehrerer Konten in Amazon Inspector mit AWS Organizations

Sie können Amazon Inspector verwenden, um mehrere Konten in einer Organisation zu verwalten. Amazon Inspector unterstützt zwei Ansätze für die Verwaltung mehrerer Konten:

- Delegierter Administrator für AWS Organizations Richtlinien — Bietet delegierten Administratoren zentrale Steuerung mit automatischer Aktivierung von Amazon Inspector für Unternehmenskonten in allen Regionen. Unternehmensrichtlinien legen fest, welche Scantypen aktiviert sind, und haben Vorrang vor der Aktivierung delegierter Administrator- und Mitgliedskonten, die nicht von Richtlinien verwaltet werden.
- Delegierter Administrator ohne AWS Organizations Richtlinien — Ein Konto, das Amazon Inspector für die Organisation ohne Verwendung von Unternehmensrichtlinien verwalten soll. Der delegierte Administrator kann Amazon Inspector für Mitgliedskonten aktivieren und die Scaneinstellungen konfigurieren.

Diese Ansätze können zusammen verwendet werden. Wenn Unternehmensrichtlinien eingerichtet sind, kontrollieren sie die Aktivierung der Ressourcentypen (welche Scantypen aktiviert sind), während delegierte Administratoren die Kontrolle über die Scan-Konfigurationseinstellungen wie Scanmodi und Deep-Inspection-Pfade behalten. In den folgenden Themen werden diese Verwaltungsansätze, die Benennung eines delegierten Administrators und die Verwaltung von Mitgliedskonten beschrieben.

Topics

- Grundlegendes zum delegierten Administratorkonto und Mitgliedskonto in Amazon Inspector
- Benennen eines delegierten Administratorkontos für Amazon Inspector

Grundlegendes zum delegierten Administratorkonto und Mitgliedskonto in Amazon Inspector

Wenn Sie Amazon Inspector in einer Umgebung mit mehreren Konten verwenden, hat das delegierte Administratorkonto Zugriff auf bestimmte Metadaten. Die Metadaten umfassen Standardscans für Amazon EC2, Amazon ECR und Lambda sowie Lambda-Code-Scans. Es enthält auch Ergebnisse

von Sicherheitsüberprüfungen für Mitgliedskonten. Dieser Abschnitt enthält Informationen darüber, welche Aktionen das delegierte Administratorkonto und die Mitgliedskonten ausführen können.

Modell zur Steuerung der Unternehmensrichtlinien

Wenn AWS Organizations Richtlinien zur Aktivierung von Amazon Inspector verwendet werden, wird ein Governance-Modell durchgesetzt, das festlegt, welche Aktionen zulässig sind:

Policy-managed Ressourcen

Ressourcen, die ausdrücklich durch Unternehmensrichtlinien aktiviert oder deaktiviert wurden, können nicht von delegierten Administratoren oder Mitgliedskonten geändert werden. API-Anfragen zur Aktivierung oder Deaktivierung richtlinienverwalteter Scantypen schlagen fehl und es wird ein eindeutiger Fehler angezeigt, der darauf hinweist, dass die Ressource durch die Unternehmensrichtlinie verwaltet wird.

Non-policy-managed Ressourcen

Ressourcen, die nicht in den Unternehmensrichtlinien angegeben sind, können normalerweise von delegierten Administratoren und Mitgliedskonten mithilfe der Amazon Inspector Inspector-Konsole oder -API verwaltet werden.

Verwaltung der Scankonfiguration

Delegierte Administratoren können jederzeit Scaneinstellungen wie EC2-Scanmodi, Deep-Inspection-Pfade und ECR-Wiederholungsdauern konfigurieren, unabhängig davon, ob die Ressourcentypen richtlinienverwaltet werden. Unternehmensrichtlinien regeln nur, ob das Scannen aktiviert ist, nicht, wie es funktioniert.

Weitere Informationen zur Erstellung und Verwaltung von Amazon Inspector Inspector-Organisationsrichtlinien finden Sie in der AWS Organizations Dokumentation zu den Amazon Inspector Inspector-Richtlinien.

Delegierte Administratoraktionen

Wenn der delegierte Administrator Einstellungen auf sein Konto anwendet, werden diese Einstellungen im Allgemeinen auf alle anderen Konten in der Organisation angewendet. Der delegierte Administrator kann auch Informationen für sein eigenes Konto und jedes zugeordnete Mitglied anzeigen und abrufen. Ein delegiertes Administratorkonto von Amazon Inspector kann die folgenden Aktionen ausführen:

- Nur das AWS Organizations Verwaltungskonto kann einen delegierten Administrator benennen und entfernen.
- Wenn Sie einen delegierten Administrator benennen, müssen Sie derselben Organisation angehören wie die Mitgliedskonten, die Sie verwalten möchten.
- Den Status von Amazon Inspector für zugehörige Konten anzeigen und verwalten, einschließlich der Aktivierung und Deaktivierung von Amazon Inspector.
- Aktivieren oder deaktivieren Sie die Scantypen für alle Mitgliedskonten in der Organisation.
- Zeigen Sie aggregierte Suchdaten für die gesamte Organisation und Suchdetails für alle Mitgliedskonten innerhalb der Organisation an.
- Erstellen und verwalten Sie Unterdrückungsregeln, die für Ergebnisse aller Konten in der Organisation gelten.
- Aktivieren Sie das erweiterte Scannen von Amazon ECR für alle Mitglieder der Organisation.
- Zeigen Sie die Ressourcenabdeckung für die gesamte Organisation an.
- Definieren Sie die Dauer für automatische erneute Scans von ECR-Container-Images für alle Mitgliedskonten in der Organisation. Die Einstellung für die Scandauer des delegierten Administrators hat Vorrang vor allen Einstellungen, die das Mitgliedskonto zuvor festgelegt hat. Alle Konten in der Organisation teilen sich die Dauer der automatisierten erneuten Scans von Amazon ECR für die delegierten Administratoren. Sie können für einzelne Konten keine unterschiedlichen Dauern für erneute Scans festlegen.
- Geben Sie fünf benutzerdefinierte Pfade für Amazon Inspector Deep Inspection für Amazon EC2 an, die für alle Konten in der Organisation verwendet werden. Dies gilt zusätzlich zu den fünf benutzerdefinierten Pfaden, die ein delegierter Administrator für sein individuelles Konto festlegen kann. Weitere Informationen zur Konfiguration benutzerdefinierter Pfade für Deep Inspection finden Sie unterBenutzerdefinierte Pfade für die Tiefeninspektion mit Amazon Inspector.
- Aktivieren und deaktivieren Sie Amazon Inspector Deep Inspection für Mitgliedskonten.
- Exportieren Sie SBOMs für alle Mitgliedskonten in der Organisation.
- Stellen Sie den Amazon EC2-Scanmodus für alle Mitgliedskonten in der Organisation ein. Weitere Informationen finden Sie unter Den Scanmodus verwalten.
- Erstellen und verwalten Sie CIS-Scankonfigurationen für alle Konten in der Organisation, mit Ausnahme von Scankonfigurationen, die von Mitgliedskonten erstellt wurden.

> **Note**
>
> Wenn ein Mitgliedskonto die Organisation verlässt, kann der delegierte Administrator die von diesem Konto geplanten Scankonfigurationen nicht mehr sehen.

- Zeigen Sie die CIS-Scanergebnisse für alle Konten in der Organisation an.
- Wenn Unternehmensrichtlinien verwendet werden, können Sie Scaneinstellungen für richtlinienverwaltete Ressourcen konfigurieren, aber die richtlinienverwalteten Scantypen selbst können nicht aktiviert oder deaktiviert werden.

Aktionen für Mitgliedskonten

Ein Mitgliedskonto kann Informationen zu seinem Konto in Amazon Inspector einsehen und abrufen, während die Einstellungen für sein Konto vom delegierten Administrator verwaltet werden. Mitgliedskonten innerhalb einer Organisation können die folgenden Aktionen in Amazon Inspector ausführen:

- Aktivieren Sie Amazon Inspector für ihr eigenes Konto.
- Sehen Sie sich die Ressourcenabdeckung für ihr eigenes Konto an.
- Details zu den Ergebnissen für ihr eigenes Konto anzeigen.
- Sehen Sie sich die Einstellung für die Dauer des automatischen erneuten Scans des ECR-Container-Images für ihr eigenes Konto an.
- Geben Sie fünf benutzerdefinierte Pfade für Amazon Inspector Deep Inspection for EC2 an, die für ihr individuelles Konto verwendet werden. Diese Pfade werden zusätzlich zu allen benutzerdefinierten Pfaden gescannt, die der delegierte Administrator für die Organisation angegeben hat. Weitere Informationen zur Konfiguration von Deep-Inspection-Pfaden finden Sie unterBenutzerdefinierte Pfade für die Tiefeninspektion mit Amazon Inspector.
- Sehen Sie sich die benutzerdefinierten Pfade an, die von Ihrem delegierten Administrator für Amazon Inspector Deep Inspection festgelegt wurden.
- Exportieren Sie SBOMs für alle Ressourcen, die mit ihrem Konto verknüpft sind.
- Sehen Sie sich den Scanmodus für ihr Konto an.
- Erstellen und verwalten Sie CIS-Scankonfigurationen für ihr Konto.
- Sehen Sie sich die Ergebnisse aller CIS-Scans nach Ressourcen in ihrem Konto an, einschließlich der vom delegierten Administrator geplanten.

- Aktivieren Sie Scantypen, die nicht durch Unternehmensrichtlinien verwaltet werden. Policy-managed Suchtypen können von Mitgliedskonten nicht aktiviert oder deaktiviert werden.

> **Note**
>
> Nach der Aktivierung kann Amazon Inspector nur durch ein delegiertes Administratorkonto deaktiviert werden.

Benennen eines delegierten Administratorkontos für Amazon Inspector

Der delegierte Administrator ist ein Konto, das einen Dienst für eine Organisation verwaltet. In diesem Thema wird beschrieben, wie Sie einen delegierten Administrator für Amazon Inspector bestimmen.

Überlegungen

Bevor Sie einen delegierten Administrator benennen, beachten Sie Folgendes:

Der delegierte Administrator kann maximal 10.000 Mitglieder verwalten.

Wenn Sie mehr als 10.000 Mitgliedskonten haben, erhalten Sie eine Benachrichtigung über das Amazon CloudWatch Personal Health Dashboard und eine E-Mail an das delegierte Administratorkonto.

> **Note**
>
> Wenn Amazon Inspector durch AWS Organizations Richtlinien für Organisationen mit mehr als 10.000 Konten (bis zu 50.000) aktiviert wird, gilt die Richtlinie für alle Konten. Es werden jedoch nur 10.000 Konten mit der Amazon Inspector Inspector-Organisation verknüpft. Das heißt, der delegierte Administrator kann die Ergebnisse und den Kontostatus nur für diese 10.000 Konten in der Amazon Inspector Inspector-Konsole einsehen.

Der delegierte Administrator ist Regional.

Amazon Inspector ist ein regionaler Service. Sie müssen die Schritte des Verfahrens überall wiederholen, AWS-Region wo Sie Amazon Inspector verwenden möchten.

Eine Organisation kann nur einen delegierten Administrator haben.

Wenn Sie in einem Konto ein Konto als delegierten Administrator festlegen AWS-Region, muss dieses Konto in allen anderen Konten der delegierte Administrator sein. AWS-Regionen

Durch das Ändern eines delegierten Administrators wird Amazon Inspector für Mitgliedskonten nicht deaktiviert.

Wenn Sie einen delegierten Administrator entfernen, werden Mitgliedskonten zu eigenständigen Konten, und die Scaneinstellungen sind davon nicht betroffen.

In Ihrem AWS Unternehmen müssen alle Funktionen aktiviert sein.

Dies ist die Standardeinstellung für AWS Organizations. Falls sie nicht aktiviert ist, finden Sie weitere Informationen unter Alle Funktionen in Ihrer Organisation aktivieren.

Unternehmensrichtlinien haben Vorrang vor delegierten Administratoreinstellungen.

Wenn Ihre Organisation AWS Organizations Richtlinien verwendet, um Amazon Inspector zu aktivieren, bestimmen die Richtlinieneinstellungen, welche Scantypen aktiviert sind. Wir empfehlen, vor der Erstellung von Unternehmensrichtlinien den delegierten Administrator zu benennen, um eine konsistente Verwaltung zu gewährleisten. Weitere Informationen finden Sie unter Modell zur Steuerung der Unternehmensrichtlinien.

Erforderliche Berechtigungen zum designieren eines delegierten Administrators

Sie benötigen die Erlaubnis, Amazon Inspector zu aktivieren und einen delegierten Amazon Inspector-Administrator zu benennen. Fügen Sie am Ende Ihrer IAM-Richtlinie die folgende Erklärung hinzu, um diese Berechtigungen zu gewähren. Weitere Informationen finden Sie unter Verwaltung von IAM-Richtlinien.

```
{
    "Sid": "PermissionsForInspectorAdmin",
    "Effect": "Allow",
    "Action": [
```

```
            "inspector2:EnableDelegatedAdminAccount",
            "organizations:EnableAWSServiceAccess",
            "organizations:RegisterDelegatedAdministrator",
            "organizations:ListDelegatedAdministrators",
            "organizations:ListAWSServiceAccessForOrganization",
            "organizations:DescribeOrganizationalUnit",
            "organizations:DescribeAccount",
            "organizations:DescribeOrganization"
        ],
        "Resource": "*"
}
```

Benennen eines delegierten Administrators für Ihr AWS Organisation

Im folgenden Verfahren wird beschrieben, wie Sie einen delegierten Administrator für Ihre Organisation bestimmen. Bevor Sie das Verfahren abschließen, stellen Sie sicher, dass Sie sich in derselben Organisation befinden wie die Mitgliedskonten, die der delegierte Administrator verwalten soll.

> **Note**
>
> Sie müssen das AWS Organizations Verwaltungskonto verwenden, um dieses Verfahren abzuschließen. Nur das AWS Organizations Verwaltungskonto kann einen delegierten Administrator benennen. Für die Benennung eines delegierten Administrators sind möglicherweise Berechtigungen erforderlich. Weitere Informationen finden Sie unter Erforderliche Berechtigungen zum designieren eines delegierten Administrators.

Wenn Sie Amazon Inspector zum ersten Mal aktivieren, erstellt Amazon Inspector die serviceverknüpfte Rolle `AWSServiceRoleForAmazonInspector` für das Konto. Informationen darüber, wie Amazon Inspector serviceverknüpfte Rollen verwendet, finden Sie unterVerwenden von serviceverknüpften Rollen für Amazon Inspector.

Console

So benennen Sie einen delegierten Administrator für Amazon Inspector

1. Melden Sie sich beim AWS Organizations Verwaltungskonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.

2. Verwenden Sie den AWS-Region Selektor, um den AWS-Region Ort anzugeben, an dem Sie den delegierten Administrator benennen möchten.
3. Wählen Sie im Navigationsbereich Allgemeine Einstellungen aus.
4. Geben Sie unter Delegierter Administrator die 12-stellige ID des Administrators ein, den AWS-Konto Sie als delegierten Administrator festlegen möchten.
5. Wählen Sie Delegieren und dann erneut Delegieren aus.

Wenn Sie einen delegierten Administrator benennen, sind standardmäßig alle Scantypen für das Konto aktiviert. Wenn Sie Amazon Inspector für das AWS Organizations Verwaltungskonto aktivieren möchten, gehen Sie wie folgt vor.

Um Amazon Inspector zu aktivieren für AWS Organizations Verwaltungskonto

1. Melden Sie sich mit dem delegierten Administratorkonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus.
3. Wählen Sie unter Konten das AWS Organizations Verwaltungskonto aus und wählen Sie dann Aktivieren aus.
4. Wählen Sie aus, welche Scantypen Sie für das AWS Organizations Verwaltungskonto aktivieren möchten, und klicken Sie dann auf Senden.

API

Benennen Sie einen delegierten Administrator, der die API verwendet

- Führen Sie den EnableDelegatedAdminAccountAPI-Vorgang mit den Anmeldeinformationen AWS-Konto des Verwaltungskontos der Organizations. Sie können dazu auch den verwenden AWS Command Line Interface , indem Sie den folgenden CLI-Befehl ausführen:`aws inspector2 enable-delegated-admin-account --delegated-admin-account-id 11111111111`.

> **Note**
>
> Stellen Sie sicher, dass Sie die Konto-ID des Kontos angeben, das Sie zu einem von Amazon Inspector delegierten Administrator machen möchten.

Die Aktivierung von Amazon Inspector scannt nach Mitgliedskonten

Sie können Amazon Inspector für Mitgliedskonten in Ihrer Organisation auf verschiedene Arten aktivieren. Welche Methode Sie wählen, hängt von Ihren Governance-Anforderungen und Ihrer Organisationsstruktur ab.

AWS Organizations Richtlinien (empfohlen für zentralisierte Verwaltung)

Verwenden Sie AWS Organizations Richtlinien, um Amazon Inspector mit zentraler Steuerung automatisch unternehmensweit zu aktivieren. Dieser Ansatz gewährleistet eine konsistente Scanabdeckung und gilt automatisch für neue Konten. Eine ausführliche Anleitung finden Sie in der AWS Organizations Dokumentation zur Erstellung von Amazon Inspector Inspector-Richtlinien.

Delegierte Administratoraktivierung

Als delegierter Administrator können Sie Amazon Inspector manuell für bestimmte Mitgliedskonten oder alle Mitgliedskonten über die Amazon Inspector Inspector-Konsole oder API aktivieren. Dieser Ansatz bietet Flexibilität, wenn Unternehmensrichtlinien nicht verwendet werden.

Selbstaktivierung des Mitgliedskontos

Mitgliedskonten können Amazon Inspector für ihr eigenes Konto aktivieren, sofern dies nicht durch Unternehmensrichtlinien eingeschränkt ist. Nach der Aktivierung wird das Konto dem delegierten Administrator zugeordnet.

Aktivieren Sie das Scannen nach Mitgliedskonten

In den folgenden Verfahren wird beschrieben, wie die Suche nach Mitgliedskonten mithilfe der delegierten Administrator- und Mitgliedskontenmethoden aktiviert wird. Informationen zu den Scantypen von Amazon Inspector finden Sie unterAutomatisierte Scantypen in Amazon Inspector.

Um das Scannen automatisch für alle Mitgliedskonten zu aktivieren

1. Melden Sie sich mit den Anmeldeinformationen für das delegierte Administratorkonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mit der Regionsauswahl die Region aus, AWS-Region in der Sie das Scannen für alle Mitgliedskonten aktivieren möchten.

3. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus. Auf der Registerkarte Konten werden alle Mitgliedskonten angezeigt, die dem AWS Organizations Verwaltungskonto zugeordnet sind.
4. Wählen Sie unter Organisation das Kästchen neben Kontonummer aus. Wählen Sie dann Aktivieren, um auszuwählen, welche Scanoptionen Sie auf Mitgliedskonten anwenden möchten. Sie können die folgenden Scanarten auswählen:

   - Amazon EC2-Scannen
   - Amazon ECR-Scannen
   - Lambda-Standardscannen
   - Scannen von Lambda-Code

   - Nachdem Sie Ihre bevorzugten Scantypen ausgewählt haben, wählen Sie Speichern.

   > **Note**
   >
   > Wenn Sie mehrere Seiten mit Konten haben, müssen Sie diesen Schritt auf jeder Seite wiederholen. Sie können das Zahnradsymbol wählen, um die Anzahl der auf jeder Seite angezeigten Konten zu ändern.

5. Aktivieren Sie die Einstellung Inspector automatisch für neue Mitgliedskonten aktivieren und wählen Sie aus, welche Scanoptionen Sie auf neue Mitgliedskonten anwenden möchten, die zu Ihrer Organisation hinzugefügt wurden. Sie können die folgenden Scanarten auswählen:

   - Amazon EC2-Scannen
   - Amazon ECR-Scannen
   - Lambda-Standardscannen
   - Scannen von Lambda-Code

   - Nachdem Sie Ihre bevorzugten Scantypen ausgewählt haben, wählen Sie Aktivieren.

   > **Note**
   >
   > Die Einstellung Inspector automatisch für neue Mitgliedskonten aktivieren aktiviert Amazon Inspector für alle future Mitglieder Ihrer Organisation.

Wenn die Anzahl der Mitgliedskonten mehr als 5.000 beträgt, wird diese Einstellung automatisch deaktiviert. Wenn die Gesamtzahl der Mitgliedskonten auf weniger als 5.000 sinkt, wird die Einstellung automatisch wieder aktiviert.

6. (Empfohlen) Wiederholen Sie jeden dieser Schritte an allen Stellen AWS-Region , an denen Sie die Suche nach Mitgliedskonten aktivieren möchten.

Um das Scannen für bestimmte Mitgliedskonten zu aktivieren

1. Melden Sie sich mit den Anmeldeinformationen für das delegierte Administratorkonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mit der Regionsauswahl die Region aus, AWS-Region in der Sie das Scannen für alle Mitgliedskonten aktivieren möchten.
3. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus. Auf der Registerkarte Konten werden alle Mitgliedskonten angezeigt, die dem AWS Organizations Verwaltungskonto zugeordnet sind.
4. Wählen Sie unter Organisation das Kästchen neben jeder Mitgliedskontonummer aus, für die Sie die Suche aktivieren möchten. Wählen Sie dann Aktivieren aus, um auszuwählen, welche Scanoptionen Sie auf Mitgliedskonten anwenden möchten. Sie können die folgenden Scanarten auswählen:

   - Amazon EC2-Scannen
   - Amazon ECR-Scannen
   - Lambda-Standardscannen
   - Scannen von Lambda-Code

   - Nachdem Sie Ihre bevorzugten Scantypen ausgewählt haben, wählen Sie Speichern.

> **Note**
>
> Wenn Sie mehrere Seiten mit Konten haben, müssen Sie diesen Schritt auf jeder Seite wiederholen. Sie können das Zahnradsymbol wählen, um die Anzahl der auf jeder Seite angezeigten Konten zu ändern.

5. (Empfohlen) Wiederholen Sie jeden dieser Schritte in allen Bereichen AWS-Region , in denen Sie das Scannen für bestimmte Mitglieder aktivieren möchten.

Um das Scannen als Mitgliedskonto zu aktivieren

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Verwenden Sie die Regionsauswahl, um den AWS-Region Ort auszuwählen, an dem Sie das Scannen für alle Mitgliedskonten aktivieren möchten.
3. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus. Auf der Registerkarte Konten werden alle Mitgliedskonten angezeigt, die dem AWS Organizations Verwaltungskonto zugeordnet sind.
4. Wählen Sie unter Organisation das Kästchen neben Ihrer Kontonummer aus. Wählen Sie dann Aktivieren, um auszuwählen, welche Scanoptionen Sie anwenden möchten. Sie können die folgenden Scantypen auswählen:

   - Amazon EC2-Scannen
   - Amazon ECR-Scannen
   - Lambda-Standardscannen
   - Scannen von Lambda-Code

   - Nachdem Sie Ihre bevorzugten Scantypen ausgewählt haben, wählen Sie Speichern.
5. (Empfohlen) Wiederholen Sie diese Schritte in jeder Region, in der Sie das Scannen für Ihr Mitgliedskonto aktivieren möchten.

> **Note**
>
> Wenn Ihr AWS Organizations Verwaltungskonto über ein delegiertes Administratorkonto für Amazon Inspector verfügt, können Sie Ihr Konto als Mitgliedskonto aktivieren, um die Scandetails einzusehen.

> **Wichtig**
>
> Wenn Organisationsrichtlinien die Aktivierung von Amazon Inspector für Ihre Konten verwalten, können die delegierten Administrator- und Mitgliedskonten die richtlinienverwalteten Scantypen nicht mithilfe von Amazon Inspector Inspector-APIs ändern. enablement/disablement API-Anfragen schlagen fehl und es wird ein Fehler angezeigt, der darauf hinweist, dass die Ressource gemäß den Unternehmensrichtlinien verwaltet wird. Sie können weiterhin zusätzliche Scantypen aktivieren, die nicht von der Richtlinie verwaltet werden.

Verknüpfung von Mitgliedskonten in Amazon Inspector aufheben

Als delegierter Administrator müssen Sie möglicherweise ein Mitgliedskonto von Ihrem Konto trennen. Wenn Sie die Verknüpfung mit einem Mitgliedskonto aufheben, ist Amazon Inspector weiterhin für das Konto aktiviert und das Konto wird zu einem eigenständigen Konto. Sie sind auch nicht mehr berechtigt, Amazon Inspector für das Konto zu verwalten. Sie können Ihrem Konto jedoch jederzeit zuvor getrennte Mitgliedskonten zuordnen. In diesem Abschnitt wird beschrieben, wie Sie als delegierter Administrator die Zuordnung von Mitgliedskonten aufheben können.

> **Note**
>
> Um die Zuordnung richtlinienverwalteter Konten zu trennen, sollte diesem Konto für den Scantyp keine Amazon Inspector Inspector-Organisationsrichtlinie zugeordnet sein.

Console

So trennen Sie die Zuordnung von Mitgliedskonten mithilfe der Konsole

1. Melden Sie sich mit den Anmeldeinformationen für das delegierte Administratorkonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home
2. Verwenden Sie die Regionsauswahl, um das Land auszuwählen, für das Sie AWS-Region die Zuordnung von Mitgliedskonten aufheben möchten.
3. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus.

4. Wählen Sie unter Organisation das Kästchen neben jeder Kontonummer aus, deren Zuordnung Sie aufheben möchten.
5. Wählen Sie im Menü Aktionen die Option Konto trennen aus.

API

So trennen Sie die Zuordnung von Mitgliedskonten mithilfe der API

Führen Sie den DisassociateMemberAPI-Vorgang aus. Geben Sie in der Anfrage die Konto-IDs an, deren Zuordnung Sie aufheben möchten.

Den delegierten Administrator in Amazon Inspector entfernen

Möglicherweise müssen Sie das delegierte Administratorkonto von Amazon Inspector entfernen. Sie können dies über das AWS Organizations Verwaltungskonto tun. Wenn Sie das delegierte Administratorkonto von Amazon Inspector entfernen, ist Amazon Inspector weiterhin für das Konto und alle Mitgliedskonten aktiviert. Das delegierte Administratorkonto und alle zugehörigen Mitgliedskonten werden zu eigenständigen Konten und behalten ihre ursprünglichen Scaneinstellungen bei.

> **Note**
>
> Wenn AWS Organizations Richtlinien die Aktivierung von Amazon Inspector verwalten, hat das Entfernen des delegierten Administrators keine Auswirkungen auf die Durchsetzung der Richtlinien. Konten bleiben gemäß den Richtlinieneinstellungen der Organisation aktiviert, obwohl die Ergebnisse der Mitgliedskonten nicht mehr in einer zentralen Konsole für delegierte Administratoren sichtbar sind, bis ein neuer delegierter Administrator benannt wird.

In diesem Abschnitt wird beschrieben, wie das delegierte Administratorkonto entfernt wird.

Den delegierten Amazon Inspector-Administrator entfernen

Die folgenden Verfahren beschreiben, wie Sie den delegierten Amazon Inspector-Administrator entfernen und Mitgliedskonten mit dem delegierten Administratorkonto verknüpfen.

Informationen zum Zuweisen eines delegierten Amazon Inspector-Administrators finden Sie unter Benennen eines delegierten Administratorkontos für Amazon Inspector.

> **Note**
>
> Nachdem Sie einen delegierten Amazon Inspector-Administrator zugewiesen haben, muss der delegierte Amazon Inspector-Administrator die Mitgliedskonten manuell zuordnen.

Um den delegierten Administrator zu entfernen

1. Melden Sie sich AWS-Managementkonsole mit dem AWS Organizations Verwaltungskonto an.
2. Öffnen Sie die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
3. Verwenden Sie die Regionsauswahl, um den AWS-Region Ort auszuwählen, von dem Sie den delegierten Administrator entfernen möchten.
4. Wählen Sie im Navigationsbereich Allgemeine Einstellungen aus.
5. Wählen Sie unter Delegierter Administrator die Option Entfernen aus, und bestätigen Sie dann Ihre Aktion.

Um Mitglieder einem neuen delegierten Administrator zuzuordnen

1. Melden Sie sich mit den Anmeldeinformationen für das delegierte Administratorkonto an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Verwenden Sie die Regionsauswahl, um das Land auszuwählen AWS-Region , dem Sie Mitglieder zuordnen möchten.
3. Wählen Sie im Navigationsbereich die Option Kontoverwaltung aus.
4. Wählen Sie unter Organisation das Kästchen neben Kontonummer aus.
5. Wählen Sie Aktionen und dann Mitglied hinzufügen aus.

Taggen von Amazon Inspector Inspector-Ressourcen

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource hinzufügen. Mithilfe von Tags können Sie AWS Ressourcen anhand bestimmter Kriterien kategorisieren. Tags bestehen aus einem Schlüssel-Wert-Paar. Der Tag-Schlüssel ist eine allgemeine Bezeichnung. Der Tag-Wert ist eine Beschreibung des Tag-Schlüssels. Mit Amazon Inspector können Sie Unterdrückungsregeln und CIS-Scankonfigurationen kennzeichnen. Sie können jeder Ihrer Amazon Inspector Inspector-Ressourcen bis zu 50 Tags hinzufügen.

Grundlagen des Kennzeichnens

Ein Tag besteht aus einem Schlüssel-Wert-Paar. Der Tag-Schlüssel ist eine allgemeine Bezeichnung. Der Tag-Wert ist eine Beschreibung des Tag-Schlüssels. In diesem Thema werden die Grundlagen der Kennzeichnung von Amazon Inspector Inspector-Ressourcen beschrieben. Beachten Sie beim Taggen von Amazon Inspector Inspector-Ressourcen Folgendes:

- Sie können Regeln zur Unterdrückung von Tags und CIS-Scankonfigurationen kennzeichnen.
- Sie können jeder Ihrer Amazon Inspector Inspector-Ressourcen bis zu 50 Tags hinzufügen.
- Tag-Schlüssel müssen eindeutig sein.
- Ein Tag-Schlüssel kann nur einen Tag-Wert haben.
- Tag-Schlüssel und Tag-Werte können maximal 128 UTF-8-Zeichen enthalten. Bei den Zeichen kann es sich um Buchstaben, Zahlen, Leerzeichen oder die folgenden Symbole handeln:. `_ . : / = + - @`
- Sie können das `aws` Präfix in keinem Ihrer Tags verwenden oder Tags mit diesem Präfix ändern. Tags mit dem `aws` Präfix sind für die Verwendung durch reserviert AWS.
- Tags, die einer Amazon Inspector Inspector-Ressource zugewiesen wurden, sind nur in Ihrem AWS Konto und dort verfügbar AWS-Region , wo Sie sie erstellt haben.
- Wenn Sie eine Ressource löschen, werden auch alle damit verknüpften Tags gelöscht.

Weitere Informationen zu Tags finden Sie unter Bewährte Methoden und Strategien im Tagging AWS Resources and Tag Editor User Guide.

> **Note**
>
> Tags sind nicht dazu bestimmt, vertrauliche oder sensible Informationen zu speichern. Verwenden Sie niemals Tags, um diese Art von Daten zu speichern. Auf Tags kann von anderen AWS Diensten aus zugegriffen werden.

Hinzufügen von -Tags

Sie können Tags zu Amazon Inspector Inspector-Ressourcen hinzufügen. Zu diesen Ressourcen gehören Unterdrückungsregeln und CIS-Scankonfigurationen. Mithilfe von Tags können Sie AWS Ressourcen anhand bestimmter Kriterien kategorisieren. In diesem Thema wird beschrieben, wie Tags zu Amazon Inspector Inspector-Ressourcen hinzugefügt werden.

Hinzufügen von Tags zu Amazon Inspector Inspector-Ressourcen

Sie können Unterdrückungsregeln und CIS-Scankonfigurationen kennzeichnen. Die folgenden Verfahren beschreiben, wie Sie Tags in der Konsole und mit der Amazon Inspector Inspector-API hinzufügen.

Hinzufügen von Tags in der Konsole

Sie können Tags zu Amazon Inspector Inspector-Ressourcen in der Konsole hinzufügen.

Hinzufügen von Tags zu Unterdrückungsregeln

Sie können den Unterdrückungsregeln während der Erstellung Tags hinzufügen. Weitere Informationen finden Sie unter Unterdrückungsregel erstellen.

Sie können eine Unterdrückungsregel auch bearbeiten, um Tags einzubeziehen. Weitere Informationen finden Sie unter Eine Unterdrückungsregel bearbeiten.

Hinzufügen von Tags zu einer CIS-Scankonfiguration

Sie können einer CIS-Scankonfiguration während der Erstellung Tags hinzufügen. Weitere Informationen finden Sie unter Eine CIS-Scankonfiguration erstellen.

Sie können eine CIS-Scankonfiguration auch so bearbeiten, dass sie Tags enthält. Weitere Informationen finden Sie unter Bearbeiten einer CIS-Scankonfiguration.

Hinzufügen von Tags mit der Amazon Inspector API

Mit der Amazon Inspector-API können Sie Amazon Inspector-Ressourcen Tags hinzufügen.

Hinzufügen von Tags zu Amazon Inspector Inspector-Ressourcen

Verwenden Sie die `TagResource` API, um Tags zu Amazon Inspector Inspector-Ressourcen hinzuzufügen. Sie müssen den ARN der Ressource und das Schlüssel-Wert-Paar für das Tag in den Befehl aufnehmen. Der folgende Beispielbefehl verwendet einen leeren Ressourcen-ARN für einen Unterdrückungsfilter. Der Schlüssel ist `CostAllocation` und der Wert ist`dev`. Informationen zu Ressourcentypen für Amazon Inspector finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Inspector2 in der Service Authorization Reference.

```
aws inspector2 tag-resource \
--resource-arn "arn:${Partition}:inspector2:${Region}:${Account}:owner/${OwnerId}/
filter/${FilterId}" \
--tags CostAllocation=dev \
--region us-west-2
```

Hinzufügen von Tags zu Unterdrückungsregeln während der Erstellung

Verwenden Sie die `CreateFilter` API, um einer Unterdrückungsregel während der Erstellung Tags hinzuzufügen.

```
aws inspector2 create-filter \
--name "ExampleSuppressionRuleECR" \
--action SUPPRESS \
--filter-criteria 'resourceType=[{comparison="EQUALS", value="AWS_ECR_IMAGE"}]' \
--tags Owner=ApplicationSecurity \
--region us-west-2
```

Hinzufügen von Tags zu einer CIS-Scankonfiguration

Verwenden Sie die `CreateCisScanConfiguration` API, um einer CIS-Scankonfiguration ein Tag hinzuzufügen.

```
aws inspector2 create-cis-scan-configuration \
--scan-name "CreateConfigWithTagsSample" \
--security-level LEVEL_2 \
--targets accountIds=SELF,targetResourceTags={InspectorCisScan=True} \
--schedule 'daily={startTime={timeOfDay=11:10,timezone=UTC}}' \
```

```
--tags Owner=SecurityEngineering \
--region us-west-2
```

Entfernen von Tags

Sie können Tags aus den Ressourcen von Amazon Inspector entfernen. Zu diesen Ressourcen gehören Unterdrückungsregeln und CIS-Scankonfigurationen. Mithilfe von Tags können Sie AWS Ressourcen anhand bestimmter Kriterien kategorisieren. In diesem Thema wird beschrieben, wie Sie Tags aus den Amazon Inspector Inspector-Ressourcen entfernen.

Tags aus Amazon Inspector Inspector-Ressourcen entfernen

Sie können Tags aus Unterdrückungsregeln und CIS-Scankonfigurationen entfernen. Die folgenden Verfahren beschreiben, wie Sie Tags in der Konsole und mit der Amazon Inspector Inspector-API entfernen.

Entfernen von Tags in der Konsole

Sie können Tags aus den Amazon Inspector Inspector-Ressourcen in der Konsole entfernen.

Tags aus den Unterdrückungsregeln entfernen

Sie können ein Tag aus einer Unterdrückungsregel entfernen, indem Sie die Unterdrückungsregel so bearbeiten, dass das Tag nicht mehr enthalten ist. Weitere Informationen finden Sie unter Eine Unterdrückungsregel bearbeiten.

Tags aus einer CIS-Scankonfiguration entfernen

Sie können ein Tag aus einer CIS-Scankonfiguration entfernen, indem Sie die CIS-Scankonfiguration so bearbeiten, dass das Tag nicht mehr enthalten ist. Weitere Informationen finden Sie unter Bearbeiten einer CIS-Scankonfiguration.

Entfernen von Tags mit der Amazon Inspector API

Mit der Amazon Inspector-API können Sie ein Tag aus einer Amazon Inspector Inspector-Ressource entfernen.

Tags aus Amazon Inspector Inspector-Ressourcen entfernen

Verwenden Sie die `UntagResource` API, um Tags aus den Amazon Inspector Inspector-Ressourcen zu entfernen.

Das folgende Snippet zeigt ein Beispiel dafür, wie Sie Tags mithilfe von einer Amazon Inspector Inspector-Ressource entfernen können. `UntagResource` Sie müssen den ARN der Ressource und den Schlüssel für das Tag in den Befehl aufnehmen. Im folgenden Beispiel wird ein leerer Ressourcen-ARN für einen Unterdrückungsfilter verwendet. Der Schlüssel lautet `CostAllocation`. Informationen zu Ressourcentypen für Amazon Inspector finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Inspector2 in der Service Authorization Reference.

```
aws inspector2 untag-resource \
--resource-arn "arn:${Partition}:inspector2:${Region}:${Account}:owner/${OwnerId}/cis-configuration/${CISScanConfigurationId}" \
--tag-keys CostAllocation \
--region us-west-2
```

Überwachung von Nutzung und Kosten in Amazon Inspector

Sie können die Amazon Inspector-Konsole und die API verwenden, um die monatlichen Amazon Inspector-Kosten für Ihre Umgebung zu prognostizieren. Wenn Sie der Amazon Inspector-Administrator für eine Umgebung mit mehreren Konten sind, können Sie die Gesamtkosten für Ihre Umgebung und die Kostenkennzahlen für alle Mitgliedskonten einsehen. In diesem Abschnitt wird beschrieben, wie Sie auf Nutzungsstatistiken zugreifen und die Nutzungskosten berechnen.

Verwenden der Nutzungskonsole

Sie können die Nutzung und die voraussichtlichen Kosten für Amazon Inspector von der Konsole aus beurteilen.

Um auf Nutzungsstatistiken zuzugreifen

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mithilfe der AWS-Region Auswahl in der oberen rechten Ecke der Seite die Region aus, in der Sie die Kosten überwachen möchten.
3. Wählen Sie im Navigationsbereich Benutzer.

Auf der Registerkarte „Nach Konto“ werden die voraussichtlichen Gesamtkosten für den Zeitraum von 30 Tagen angezeigt, der unter Kontonutzung aufgeführt ist. Wählen Sie in der Tabelle unter der Spalte „Voraussichtliche Kosten“ einen Wert aus, um eine Aufschlüsselung der Nutzung nach Scan-Typ für dieses Konto anzuzeigen. In diesem Detailbereich können Sie auch sehen, für welche Scantypen eine kostenlose Testversion für dieses Konto aktiv ist.

Wenn Sie der delegierte Administrator für eine Organisation sind, wird in der Tabelle für jedes Konto innerhalb Ihrer Organisation eine Zeile angezeigt. Wenn die Verknüpfung eines Kontos in Ihrer Organisation aufgehoben wird, zeigt die Konsole die voraussichtlichen Kosten als - an.

Auf der Registerkarte „Nach Scan-Typ“ finden Sie eine Aufschlüsselung der tatsächlichen Nutzung im aktuellen Zeitraum von 30 Tagen nach Scantyp. Dies sind die Informationen, die zur Berechnung der voraussichtlichen Kosten auf der Registerkarte „Nach Konto“ verwendet werden.

Wenn Sie der delegierte Administrator einer Organisation sind, können Sie die Nutzung für jedes Konto in Ihrer Organisation einsehen.

Auf dieser Registerkarte können Sie jeden der folgenden Bereiche für Nutzungsstatistiken erweitern:

Amazon EC2-Scannen

Die Amazon Inspector-Nutzungskonsole verfolgt die folgenden Kennzahlen für agentengestütztes Scannen und agentenloses Scannen:

- Instances (Avg) — Amazon Inspector berechnet anhand der Abdeckungsstunden die durchschnittliche Anzahl der Ressourcen für das Scannen von EC2-Instances. Der Durchschnitt ist die Gesamtzahl der Abdeckungsstunden geteilt durch 720 Stunden (die Anzahl der Stunden in einem Zeitraum von 30 Tagen).
- Servicestunden — Für Amazon EC2-Scans ist dies die Gesamtzahl der Stunden innerhalb der letzten 30 Tage, in denen Amazon Inspector aktiven Schutz für jede EC2-Instance in einem Konto bereitgestellt hat. Bei EC2-Instances sind die Servicezeiten die Stunden zwischen dem Zeitpunkt, an dem Amazon Inspector die Instance entdeckt hat, bis sie beendet oder gestoppt oder anhand von Tags von den Scans ausgeschlossen wurde. (wenn Sie eine gestoppte Instance neu starten oder ein Ausschluss-Tag entfernen, nimmt Amazon Inspector den Schutz wieder auf und die Servicezeiten für diese Instance fallen weiterhin an).

CIS-Instance-Scans — Die Gesamtzahl der CIS-Scans, die für Instances im Konto durchgeführt wurden.

Amazon ECR-Scannen

Erste Scans — Die Summe der ersten Scans von Bildern auf dem Konto innerhalb der letzten 30 Tage.

Erneute Scans — Die Gesamtzahl der Neuscans von Bildern im Konto innerhalb der letzten 30 Tage. Ein erneuter Scan ist ein Scan, der an einem ECR-Bild durchgeführt wird, das Amazon Inspector zuvor gescannt hat. Wenn Sie Ihr ECR-Repository für kontinuierliches Scannen konfiguriert haben, erfolgen erneute Scans automatisch, wenn Amazon Inspector seiner Datenbank ein neues Common Vulnerabilities and Exposures (CVE) hinzufügt.

Lambda-Scannen

Die Amazon Inspector-Nutzungskonsole verfolgt die folgenden Metriken für das Lambda-Standardscannen und das Lambda-Code-Scannen:

- Anzahl der Lambda-Funktionen (Avg) — Amazon Inspector berechnet anhand der Empfangsstunden die durchschnittliche Anzahl der Funktionen für das Scannen von Lambda-Funktionen. Der Durchschnitt ist die Gesamtzahl der Abdeckungsstunden geteilt durch 720 Stunden (die Anzahl der Stunden in einem Zeitraum von 30 Tagen).

- Servicezeiten — Für das Scannen von Lambda-Funktionen ist dies die Gesamtzahl der Stunden innerhalb der letzten 30 Tage, in denen Amazon Inspector für jede Lambda-Funktion in einem Konto eine aktive Abdeckung bereitgestellt hat. Für AWS Lambda Funktionen werden die Servicezeiten ab dem Zeitpunkt berechnet, an dem Amazon Inspector eine Funktion entdeckt, bis zu dem Zeitpunkt, zu dem sie gelöscht oder von den Scans ausgeschlossen wird. Wenn eine ausgeschlossene Funktion erneut enthalten ist, fallen die Servicezeiten für diese Funktion weiterhin an.

Scannen virtueller Azure-Maschinen

Die Amazon Inspector-Nutzungskonsole verfolgt die folgenden Metriken für das Scannen von Azure-VMs:

- Anzahl der Azure-VMs (Avg) — Die durchschnittliche Anzahl der Azure-VMs, die vom Amazon Inspector VM Scanner während des Zeitraums von 30 Tagen überwacht wurden.
- Servicestunden — Die Gesamtzahl der Stunden, in denen Amazon Inspector aktiv nach Sicherheitslücken für jede Azure-VM in einem Konto gesucht hat.

Scannen von Azure Function Apps

Die Amazon Inspector-Nutzungskonsole verfolgt die folgenden Metriken für das Scannen der Azure Function App:

- Anzahl der Function Apps (Avg) — Die durchschnittliche Anzahl der Azure Function Apps, die während des Zeitraums von 30 Tagen überwacht wurden.
- Servicestunden — Die Gesamtzahl der Stunden, in denen Amazon Inspector aktive Codeabhängigkeitsscans für jede Azure Function-App in einem Konto bereitgestellt hat.

Scannen von Bildern in Azure Container Registry (ACR)

Die Amazon Inspector-Nutzungskonsole verfolgt die folgenden Metriken für das Scannen von Azure Container Registry-Images:

- Erste Scans — Die Gesamtzahl der ersten Scans von Azure-Container-Images innerhalb der letzten 30 Tage.
- Erneute Scans — Die Gesamtzahl der erneuten Scans für Azure-Container-Images innerhalb der letzten 30 Tage. Ein erneuter Scan erfolgt automatisch, wenn Amazon Inspector seiner Datenbank ein neues CVE hinzufügt, das sich auf ein zuvor gescanntes Bild auswirkt.

> **Note**
>
> Jeder Azure-Ressourcentyp beinhaltet eine kostenlose 30-Tage-Testversion. Die kostenlose Testversion beginnt unabhängig für jeden Ressourcentyp, wenn Amazon Inspector beginnt, diesen Ressourcentyp in Ihrem Konto zu scannen.

Erfahren Sie, wie Amazon Inspector die Nutzungskosten berechnet

Bei den von Amazon Inspector angegebenen Kosten handelt es sich um Schätzungen, nicht um tatsächliche Kosten. Sie können daher von denen in Ihrer AWS Billing Konsole abweichen.

Beachten Sie auf der Seite „Nutzung“ die folgenden Hinweise zur Berechnung der Kosten durch Amazon Inspector:

- Die Nutzungskosten beziehen sich nur auf die aktuelle Region. Die Preise pro Scan-Typ variieren je nach AWS Region. Die genauen Preise pro Region finden Sie unter Preise für Amazon Inspector
- Alle Nutzungsprognosen sind auf den nächsten US-Dollar gerundet.
- Rabatte sind nicht in den voraussichtlichen Kosten enthalten.
- Die voraussichtlichen Kosten stellen die Gesamtkosten für den 30-tägigen Nutzungszeitraum pro Scan-Typ dar. Wenn ein Konto weniger als 30 Tage genutzt wurde, prognostiziert Amazon Inspector die Kosten nach 30 Tagen so, als ob alle derzeit abgedeckten Ressourcen für den Rest des 30-Tage-Zeitraums abgedeckt bleiben würden.
- Die Kosten pro Scan-Typ werden auf der Grundlage der folgenden Kriterien berechnet:
  - EC2-Scannen: Die Kosten spiegeln die durchschnittliche Anzahl der EC2-Instances wider, die Amazon Inspector in den letzten 30 Tagen abgedeckt hat.
  - Scannen von ECR-Containern: Die Kosten entsprechen der Summe aus der Anzahl der ersten Bildscans und der erneuten Bildscans in den letzten 30 Tagen.
  - Lambda-Standardscanning: Die Kosten spiegeln die durchschnittliche Anzahl der Lambda-Funktionen wider, die Amazon Inspector in den letzten 30 Tagen abdeckt hat.
  - Lambda-Code-Scannen: Die Kosten spiegeln die durchschnittliche Anzahl der Lambda-Funktionen wider, die Amazon Inspector in den letzten 30 Tagen abgedeckt hat.

Über die kostenlose Amazon Inspector-Testversion

In Amazon Inspector gibt es für jeden Scan-Typ eine kostenlose Testversion. Wenn Sie einen Scan-Typ aktivieren, registrieren Sie sich automatisch für eine kostenlose 15-Tage-Testversion für diesen Scan-Typ. Sobald die kostenlose Testversion beginnt, läuft sie automatisch nach 15 Tagen ab, auch wenn Sie den Scan-Typ deaktivieren.

> **Note**
>
> Die kostenlose Testversion gilt nicht für CIS-Scannen.

Sicherheit in Amazon Inspector

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-party Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für Amazon Inspector gelten, finden Sie unter AWS Services im Bereich nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amazon Inspector anwenden können. In den folgenden Themen erfahren Sie, wie Sie Amazon Inspector konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amazon Inspector Inspector-Ressourcen überwachen und sichern können.

Topics

- Datenschutz in Amazon Inspector
- Identity and Access Management für Amazon Inspector
- Überwachung von Amazon Inspector
- Konformitätsvalidierung für Amazon Inspector
- Resilienz in Amazon Inspector
- Infrastruktursicherheit in Amazon Inspector
- Reaktion auf Vorfälle in Amazon Inspector
- Sicherheitsforschung von Amazon Inspector

- Greifen Sie über einen Schnittstellenendpunkt auf Amazon Inspector zu (AWS PrivateLink)

Datenschutz in Amazon Inspector

Das AWS Modell der gilt für den Datenschutz in Amazon Inspector. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz . Weitere Informationen zum Datenschutz in Europa finden Sie im Zentrum für die Datenschutz-Grundverordnung (DSGVO).

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon Inspector oder anderen AWS-Services über die Konsole AWS CLI, API oder AWS SDKs

arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Themen

- Verschlüsselung im Ruhezustand
- Verschlüsselung während der Übertragung

Verschlüsselung im Ruhezustand

Standardmäßig speichert Amazon Inspector Daten im Ruhezustand mithilfe von AWS Verschlüsselungslösungen. Amazon Inspector verschlüsselt Daten wie die folgenden:

- Der Ressourcenbestand wurde mit AWS Systems Manager erfasst.
- Aus Amazon Elastic Container Registry-Images analysierter Ressourcenbestand
- Generierte Sicherheitsergebnisse unter Verwendung AWS eigener Verschlüsselungsschlüssel von AWS Key Management Service

Sie können AWS eigene Schlüssel nicht verwalten, verwenden oder anzeigen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme ändern, um Schlüssel zu schützen, die Ihre Daten verschlüsseln. Weitere Informationen finden Sie unter AWS -eigene Schlüssel.

Wenn Sie Amazon Inspector deaktivieren, werden alle Ressourcen, die es für Sie speichert oder verwaltet, dauerhaft gelöscht, z. B. gesammeltes Inventar und Sicherheitserkenntnisse.

Verschlüsselung im Ruhezustand für den Code in Ihren Ergebnissen

Beim Scannen von Amazon Inspector Lambda-Code arbeitet Amazon Inspector mit Amazon Q zusammen, um Ihren Code auf Sicherheitslücken zu scannen. Wenn eine Sicherheitslücke erkannt wird, extrahiert Amazon Q einen Codeausschnitt, der die Sicherheitslücke enthält, und speichert diesen Code, bis Amazon Inspector Zugriff anfordert. Standardmäßig verwendet Amazon Q einen AWS eigenen Schlüssel, um den extrahierten Code zu verschlüsseln. Sie können Amazon Inspector jedoch so konfigurieren, dass Ihr eigener, vom Kunden verwalteter AWS KMS Schlüssel für die Verschlüsselung verwendet wird.

Der folgende Arbeitsablauf erklärt, wie Amazon Inspector den von Ihnen konfigurierten Schlüssel verwendet, um Ihren Code zu verschlüsseln:

1. Sie stellen Amazon Inspector mithilfe der Amazon Inspector UpdateEncryptionKeyInspector-API einen AWS KMS Schlüssel zur Verfügung.
2. Amazon Inspector leitet die Informationen über Ihren AWS KMS Schlüssel an Amazon Q weiter, und Amazon Q speichert die Informationen für die future Verwendung.
3. Amazon Q verwendet den KMS-Schlüssel, den Sie in Amazon Inspector über die Schlüsselrichtlinie konfiguriert haben.
4. Amazon Q erstellt aus Ihrem Schlüssel einen verschlüsselten AWS KMS Datenschlüssel und speichert ihn. Dieser Datenschlüssel wird verwendet, um Ihre von Amazon Q gespeicherten Codedaten zu verschlüsseln.
5. Wenn Amazon Inspector Daten aus Codescans anfordert, verwendet Amazon Q den KMS-Schlüssel, um den Datenschlüssel zu entschlüsseln. Wenn Sie das Lambda-Code-Scannen deaktivieren, löscht Amazon Q den zugehörigen Datenschlüssel.

Berechtigungen für die Codeverschlüsselung mit einem vom Kunden verwalteten Schlüssel

Für die Verschlüsselung müssen Sie einen KMS-Schlüssel mit einer Richtlinie erstellen, die eine Anweisung enthält, mit der Amazon Inspector und Amazon Q die folgenden Aktionen ausführen können.

- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:Encrypt`
- `kms:GenerateDataKey`
- `kms:GenerateDataKeyWithoutPlainText`

Richtlinienanweisung

Sie können die folgende Richtlinienerklärung verwenden, wenn Sie den KMS-Schlüssel erstellen.

> **Note**
>
> Ersetzen Sie es *`account-id`* durch Ihre 12-stellige AWS-Konto ID. *`Region`*Ersetzen Sie es durch das AWS-Region , wo Sie Amazon Inspector und Lambda-Code-Scanning aktiviert haben. *`role-ARN`*Ersetzen Sie es durch den Amazon-Ressourcennamen für Ihre IAM-Rolle.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "q.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:EncryptionContext:aws:qdeveloper:lambda-codescan-scope": "account-id"
    },
    "StringEquals": {
      "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:qdeveloper:Region:account-id:scans/*"
    }
  }
},
{
  "Effect": "Allow",
  "Principal": {
    "Service": "q.amazonaws.com"
  },
  "Action": "kms:DescribeKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:qdeveloper:Region:account-id:scans/*"
    }
  }
},
{
  "Effect": "Allow",
```

```
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:GenerateDataKey"
  ],
  "Principal": {
    "AWS": "role-ARN"
  },
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "inspector2.Region.amazonaws.com"
    },
    "StringLike": {
      "kms:EncryptionContext:aws:qdeveloper:lambda-codescan-scope": "account-id"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "kms:DescribeKey"
  ],
  "Principal": {
    "AWS": "role-ARN"
  },
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "inspector2.Region.amazonaws.com"
    }
  }
}
```

Die Richtlinienerklärung ist in JSON formatiert. Nachdem Sie die Erklärung hinzugefügt haben, überprüfen Sie die Richtlinie, um sicherzustellen, dass die Syntax gültig ist. Wenn es sich bei der Aussage um die letzte Aussage in der Richtlinie handelt, setzen Sie hinter die schließende Klammer für die vorherige Anweisung ein Komma. Wenn es sich bei der Aussage um die erste Aussage oder um eine Aussage zwischen zwei bestehenden Aussagen in der Richtlinie handelt, setzen Sie hinter der schließenden Klammer für die Anweisung ein Komma.

> **Note**
>
> Amazon Inspector unterstützt keine Zuschüsse mehr zur Verschlüsselung von Codefragmenten, die aus Paketen extrahiert wurden. Wenn Sie eine auf Zuschüssen basierende Richtlinie verwenden, können Sie weiterhin auf Ihre Ergebnisse zugreifen. Wenn Sie jedoch Ihren KMS-Schlüssel aktualisieren oder zurücksetzen oder das Lambda-Code-Scannen deaktivieren, müssen Sie die in diesem Abschnitt beschriebene KMS-Schlüsselrichtlinie verwenden.

Wenn Sie den Verschlüsselungsschlüssel für Ihr Konto einrichten, aktualisieren oder zurücksetzen, müssen Sie eine Amazon Inspector-Administratorrichtlinie verwenden, z. B. die AWS verwaltete Richtlinie`AmazonInspector2FullAccess`.

Konfiguration der Verschlüsselung mit einem vom Kunden verwalteten Schlüssel

Um die Verschlüsselung für Ihr Konto mit einem vom Kunden verwalteten Schlüssel zu konfigurieren, müssen Sie ein Amazon Inspector-Administrator mit den unter beschriebenen Berechtigungen seinBerechtigungen für die Codeverschlüsselung mit einem vom Kunden verwalteten Schlüssel. Darüber hinaus benötigen Sie einen AWS KMS Schlüssel in derselben AWS Region wie Ihre Ergebnisse oder einen Schlüssel für mehrere Regionen. Sie können einen vorhandenen symmetrischen Schlüssel in Ihrem Konto verwenden oder mithilfe der AWS Managementkonsole oder der APIs einen symmetrischen, vom Kunden verwalteten Schlüssel erstellen. AWS KMS Weitere Informationen finden Sie im Benutzerhandbuch unter Erstellen symmetrischer AWS KMSAWS KMS Verschlüsselungsschlüssel.

> **Note**
>
> Ab dem 13. Juni 2025 ändert sich der Dienstprinzipal bei AWS KMS Anfragen, die CloudTrail während des encryption/decryption Codeausschnitts angemeldet wurden, von „codeguru-reviewer“ zu „q“.

Verwenden der Amazon Inspector API zur Konfiguration der Verschlüsselung

Um einen Schlüssel für die Verschlüsselung für den UpdateEncryptionKeyBetrieb der Amazon Inspector-API festzulegen, während Sie als Amazon Inspector-Administrator angemeldet sind. Verwenden Sie in der API-Anfrage das `kmsKeyId` Feld, um den ARN des AWS KMS Schlüssels

anzugeben, den Sie verwenden möchten. Für `scanType` Enter `CODE` und für `resourceType` Enter`AWS_LAMBDA_FUNCTION`.

Sie können die UpdateEncryptionKeyAPI verwenden, um zu überprüfen, welchen AWS KMS Schlüssel Amazon Inspector für die Verschlüsselung verwendet.

> **Note**
>
> Wenn Sie versuchen zu verwenden, `GetEncryptionKey` obwohl Sie keinen vom Kunden verwalteten Schlüssel eingerichtet haben, gibt der Vorgang einen `ResourceNotFoundException` Fehler zurück, was bedeutet, dass ein AWS eigener Schlüssel für die Verschlüsselung verwendet wird.

Wenn Sie den Schlüssel löschen oder seine Richtlinie dahingehend ändern, dass der Zugriff auf Amazon Inspector oder Amazon Q verweigert wird, können Sie nicht mehr auf Ihre gefundenen Sicherheitslücken zugreifen und der Lambda-Code-Scan schlägt für Ihr Konto fehl.

Sie können `ResetEncryptionKey` damit fortfahren, einen AWS eigenen Schlüssel zur Verschlüsselung von Code zu verwenden, der im Rahmen Ihrer Amazon Inspector Inspector-Ergebnisse extrahiert wurde.

Verschlüsselung während der Übertragung

AWS verschlüsselt alle Daten, die zwischen AWS internen Systemen und anderen AWS Diensten übertragen werden. AWS Systems Manager sammelt Telemetriedaten von kundeneigenen EC2-Instances, an die es zur Bewertung AWS über einen durch Transport Layer Security (TLS) geschützten Kanal sendet. Die Ergebnisse der Amazon ECR- und AWS Lambda-Funktionsscans, die an Security Hub CSPM gesendet werden, werden über einen Kanal verschlüsselt. TLS-protected Weitere Informationen finden Sie unter Datenschutz in Systems Manager, um zu erfahren, wie SSM Daten bei der Übertragung verschlüsselt.

Identity and Access Management für Amazon Inspector

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert

(angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon Inspector Inspector-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- So arbeitet Amazon Inspector mit IAM
- Identity-based Richtlinienbeispiele für Amazon Inspector
- AWS verwaltete Richtlinien für Amazon Inspector
- Verwenden von serviceverknüpften Rollen für Amazon Inspector
- Fehlerbehebung bei Identität und Zugriff auf Amazon Inspector

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung bei Identität und Zugriff auf Amazon Inspector).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe So arbeitet Amazon Inspector mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Identity-based Richtlinienbeispiele für Amazon Inspector).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden

finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Richtlinien

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identity-based Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein.

Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Resource-based Richtlinien

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Resource-based Richtlinien sind Inline-Richtlinien, die sich in diesem Dienst befinden. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Ressourcen-Kontrollrichtlinien (RCPs) – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter Ressourcen-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

So arbeitet Amazon Inspector mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon Inspector zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen mit Amazon Inspector verwendet werden können.

IAM-Funktionen, die Sie mit Amazon Inspector verwenden können

IAM-Feature	Unterstützung für Amazon Inspector
Identity-based Richtlinien	Ja
Resource-based Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Nein
Service-linked Rollen	Ja

Einen allgemeinen Überblick darüber, wie Amazon Inspector und andere AWS-Services mit den meisten IAM-Funktionen funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch.

Identity-based Richtlinien für Amazon Inspector

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität anhängen können, z. B. an einen IAM-Benutzer, eine Benutzergruppe oder eine Rolle.

Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Identity-based Richtlinienbeispiele für Amazon Inspector

Beispiele für identitätsbasierte Richtlinien von Amazon Inspector finden Sie unter. Identity-based Richtlinienbeispiele für Amazon Inspector

Resource-based Richtlinien innerhalb von Amazon Inspector

Unterstützt ressourcenbasierte Richtlinien: Nein

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Politische Maßnahmen für Amazon Inspector

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der Amazon Inspector-Aktionen finden Sie unter Von Amazon Inspector definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in Amazon Inspector verwenden das folgende Präfix vor der Aktion:

```
inspector2
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "inspector2:action1",
      "inspector2:action2"
         ]
```

Beispiele für identitätsbasierte Richtlinien von Amazon Inspector finden Sie unter. Identity-based Richtlinienbeispiele für Amazon Inspector

Richtlinienressourcen für Amazon Inspector

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der Amazon Inspector-Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon Inspector definierte Ressourcen in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon Inspector definierte Aktionen.

Beispiele für identitätsbasierte Richtlinien von Amazon Inspector finden Sie unter. Identity-based Richtlinienbeispiele für Amazon Inspector

Schlüssel zu den Richtlinienbedingungen für Amazon Inspector

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der Amazon Inspector-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Inspector in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon Inspector definierte Aktionen.

Beispiele für identitätsbasierte Richtlinien von Amazon Inspector finden Sie unter. Identity-based Richtlinienbeispiele für Amazon Inspector

ACLs in Amazon Inspector

Unterstützt ACLs: Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon Inspector

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Attribute-based Access Control (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert, die als Tags bezeichnet werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen mit Amazon Inspector verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Cross-service Hauptberechtigungen für Amazon Inspector

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für Amazon Inspector

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Amazon Inspector beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon Inspector Sie dazu anleitet.

Service-linked Rollen für Amazon Inspector

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer AWS-Service verknüpft ist. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen auszuführen. Service-linked Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS-Services , die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Dienst, der einen `Yes` in der Service-linked Rollenspalte enthält. Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Identity-based Richtlinienbeispiele für Amazon Inspector

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon Inspector Inspector-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amazon Inspector definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Inspector in der Service Authorization Reference.

Themen

- Best Practices für Richtlinien
- Verwenden der Amazon Inspector Inspector-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Nur-Lese-Zugriff auf alle Amazon Inspector Inspector-Ressourcen zulassen
- Vollzugriff auf alle Amazon Inspector Inspector-Ressourcen zulassen

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand Amazon Inspector Inspector-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn Sie identitätsbasierte Richtlinien erstellen oder bearbeiten, befolgen Sie diese Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben,

um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon Inspector Inspector-Konsole

Um auf die Amazon Inspector Inspector-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon Inspector Inspector-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die Amazon Inspector-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch den Amazon Inspector `ConsoleAccess` oder die `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Nur-Lese-Zugriff auf alle Amazon Inspector Inspector-Ressourcen zulassen

Dieses Beispiel zeigt eine Richtlinie, die nur Lesezugriff auf alle Amazon Inspector Inspector-Ressourcen ermöglicht.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "inspector2:Describe*",
                "inspector2:Get*",
                "inspector2:BatchGet*",
                "inspector2:List*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:ListDelegatedAdministrators",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

Vollzugriff auf alle Amazon Inspector Inspector-Ressourcen zulassen

Dieses Beispiel zeigt eine Richtlinie, die vollen Zugriff auf alle Amazon Inspector Inspector-Ressourcen ermöglicht.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "inspector2:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "inspector2.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:EnableAWSServiceAccess",
                "organizations:RegisterDelegatedAdministrator",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete Richtlinien für Amazon Inspector

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AmazonInspector2FullAccess_v2

Sie können die `AmazonInspector2FullAccess_v2`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt vollen Zugriff auf Amazon Inspector und Zugriff auf andere verwandte Dienste.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `inspector2`— Ermöglicht den vollständigen Zugriff auf Amazon Inspector Inspector-APIs.
- `codeguru-security`— Ermöglicht Administratoren das Abrufen von Sicherheitsergebnissen und Konfigurationseinstellungen für ein Konto.
- `iam`— Ermöglicht Amazon Inspector, die serviceverknüpften Rollen `AWSServiceRoleForAmazonInspector2AWSServiceRoleForAmazonInspector2Agentless`, und `AWSServiceRoleForAmazonInspector2ThirdParty` zu erstellen.

`AWSServiceRoleForAmazonInspector2`ist erforderlich, damit Amazon Inspector Vorgänge wie das Abrufen von Informationen über Amazon EC2 EC2-Instances, Amazon ECR-Repositorys und Amazon ECR-Container-Images ausführen kann. Es ist auch erforderlich, mit Schlüsseln verschlüsselte Amazon EBS-Snapshots zu entschlüsseln. AWS KMS Darüber hinaus ermöglicht es die Erstellung der `AWSServiceRoleForConfigThirdParty` serviceverknüpften Rolle, die für die Aufzeichnung von Multi-Cloud-Ressourcen erforderlich ist. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon Inspector.

- `config`— Ermöglicht Amazon Inspector, Konnektoren für die Multi-Cloud-Ressourcenerkennung zu erstellen, zu lesen und aufzulisten.
- `organizations`— `AllowServicePrincipalBasedAccessToOrganizationApis` ermöglicht nur Service Principals, servicebasierte Rollen für eine Organisation zu erstellen AWS-Konten, sie AWS-Konto als delegierten Administrator für eine Organisation zu registrieren und delegierte Administratoren in einer Organisation aufzulisten. `AllowOrganizationalBasedAccessToOrganizationApis`ermöglicht es dem Versicherungsnehmer, Informationen, insbesondere ARNs auf Ressourcenebene, über eine Organisationseinheit abzurufen. `AllowAccountsBasedAccessToOrganizationApis`ermöglicht es dem Versicherungsnehmer, Informationen, insbesondere ARNs auf Ressourcenebene, über eine abzurufen. AWS-Konto`AllowAccessToOrganizationApis`ermöglicht es dem Versicherungsnehmer, AWS-Services integrierte Informationen zu einer Organisation und Organisation einzusehen. Die Richtlinie ermöglicht das Auflisten von Inspector-Organisationsrichtlinien mit Filterung nach Inspector-Richtlinientypen, das Anzeigen von Delegierungsressourcenrichtlinien, die von Verwaltungskonten eingerichtet wurden, und das Anzeigen der effektiven Inspector-Richtlinien, die auf Konten angewendet wurden.

> **Note**
>
> Amazon Inspector verwendet CodeGuru keine Lambda-Scans mehr. AWS wird den Support für den 20. CodeGuru November 2025 einstellen. Weitere Informationen finden Sie unter Ende des Supports für CodeGuru Security. Amazon Inspector verwendet jetzt Amazon Q zur Durchführung von Lambda-Scans und benötigt nicht die in diesem Abschnitt beschriebenen Berechtigungen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AmazonInspector2FullAccess_v2 im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSInspector2OrganizationsAccess

Sie können die `AWSInspector2OrganizationsAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Administratorberechtigungen zur Aktivierung und Verwaltung von Amazon Inspector für eine Organisation in AWS Organizations. Die Berechtigungen für diese Richtlinie ermöglichen es dem Organisationsverwaltungskonto, das delegierte Administratorkonto für Amazon Inspector zu bestimmen. Sie ermöglichen es dem delegierten Administratorkonto auch, Organisationskonten als Mitgliedskonten zu aktivieren.

Diese Richtlinie bietet nur Berechtigungen für AWS Organizations. Das Organisationsverwaltungskonto und das delegierte Administratorkonto erfordern ebenfalls Berechtigungen für zugehörige Aktionen. Diese Berechtigungen können mithilfe der `AmazonInspector2FullAccess_v2` verwalteten Richtlinie erteilt werden.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `organizations:ListAccounts`— Ermöglicht Prinzipalen das Abrufen der Liste der Konten, die Teil einer Organisation sind.
- `organizations:DescribeOrganization`— Ermöglicht Prinzipalen das Abrufen von Informationen über die Organisation.
- `organizations:ListRoots`— Ermöglicht Prinzipalen, das Stammverzeichnis einer Organisation aufzulisten.
- `organizations:ListDelegatedAdministrators`— Ermöglicht Prinzipalen, den delegierten Administrator einer Organisation aufzulisten.
- `organizations:ListAWSServiceAccessForOrganization`— Ermöglicht Prinzipalen, diejenigen aufzulisten, AWS-Services die von einer Organisation verwendet werden.
- `organizations:ListOrganizationalUnitsForParent`— Ermöglicht Prinzipalen, die untergeordneten Organisationseinheiten (OU) einer übergeordneten OU aufzulisten.
- `organizations:ListAccountsForParent`— Ermöglicht Prinzipalen, die untergeordneten Konten einer übergeordneten Organisationseinheit aufzulisten.
- `organizations:ListParents`— Führt die Stamm- oder Organisationseinheiten (OUs) auf, die der angegebenen untergeordneten Organisationseinheit oder dem angegebenen untergeordneten Konto als unmittelbare übergeordnete Einheit dienen.

- `organizations:DescribeAccount` – Ermöglicht Prinzipalen den Abruf von Informationen über ein Konto in der Organisation.
- `organizations:DescribeOrganizationalUnit`— Ermöglicht Prinzipalen das Abrufen von Informationen über eine Organisationseinheit in der Organisation.
- `organizations:ListPolicies`— Ruft die Liste aller Richtlinien in einer Organisation eines bestimmten Typs ab.
- `organizations:ListPoliciesForTarget`— Führt die Richtlinien auf, die direkt mit dem angegebenen Zielstamm, der Organisationseinheit (OU) oder dem angegebenen Konto verknüpft sind.
- `organizations:ListTargetsForPolicy`— Führt alle Stammverzeichnisse, Organisationseinheiten (OUs) und Konten auf, denen die angegebene Richtlinie zugeordnet ist.
- `organizations:DescribeResourcePolicy`— Ruft Informationen über eine Ressourcenrichtlinie ab.
- `organizations:EnableAWSServiceAccess`— Ermöglicht Prinzipalen, die Integration mit Organizations zu ermöglichen.
- `organizations:RegisterDelegatedAdministrator`— Ermöglicht es den Prinzipalen, das delegierte Administratorkonto festzulegen.
- `organizations:DeregisterDelegatedAdministrator`— Ermöglicht Prinzipalen, das delegierte Administratorkonto zu entfernen.
- `organizations:DescribePolicy`— Ruft Informationen zu einer Richtlinie ab.
- `organizations:DescribeEffectivePolicy`— Gibt den Inhalt der gültigen Richtlinie für den angegebenen Richtlinientyp und das angegebene Konto zurück.
- `organizations:CreatePolicy`— Erstellt eine Richtlinie eines bestimmten Typs, die Sie einem Stamm, einer Organisationseinheit (OU) oder einer Einzelperson zuordnen können AWS-Konto.
- `organizations:UpdatePolicy`— Aktualisiert eine bestehende Richtlinie mit einem neuen Namen, einer neuen Beschreibung oder einem neuen Inhalt.
- `organizations:DeletePolicy`— Löscht die angegebene Richtlinie aus Ihrer Organisation.
- `organizations:AttachPolicy`— Fügt eine Richtlinie einem Stammkonto, einer Organisationseinheit (OU) oder einem Einzelkonto hinzu.
- `organizations:DetachPolicy`— Trennt eine Richtlinie von einem Zielstamm, einer Organisationseinheit (OU) oder einem Konto.
- `organizations:EnablePolicyType`— Aktiviert einen Richtlinientyp in einem Stamm.

- `organizations:DisablePolicyType`— Deaktiviert einen Organisationsrichtlinientyp in einem Stamm.
- `organizations:TagResource`— Fügt einer angegebenen Ressource ein oder mehrere Tags hinzu.
- `organizations:UntagResource`— Entfernt alle Tags mit den angegebenen Schlüsseln aus einer angegebenen Ressource.
- `organizations:ListTagsForResource`— Listet Tags auf, die an eine angegebene Ressource angehängt sind.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie AWSInspector2OrganizationsAccessim Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonInspector2FullAccess

Sie können die `AmazonInspector2FullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Administratorberechtigungen, die vollen Zugriff auf Amazon Inspector ermöglichen.

> **Important**
>
> Für mehr Sicherheit und eingeschränkte Berechtigungen für Inspector 2-Dienstprinzipale empfehlen wir die Verwendung von AmazonInspector2FullAccess_v2.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `inspector2`— Ermöglicht vollen Zugriff auf die Funktionen von Amazon Inspector.
- `iam`— Ermöglicht Amazon Inspector, die serviceverknüpften Rollen `AWSServiceRoleForAmazonInspector2AWSServiceRoleForAmazonInspector2Agentless`, und `AWSServiceRoleForAmazonInspector2ThirdParty` zu erstellen. `AWSServiceRoleForAmazonInspector2`ist erforderlich, damit Amazon Inspector Vorgänge wie das Abrufen von Informationen über Ihre Amazon EC2 EC2-Instances, Amazon ECR-

Repositorys und Container-Images ausführen kann. Es ist auch erforderlich, dass Amazon Inspector Ihr VPC-Netzwerk analysiert und Konten beschreibt, die mit Ihrer Organisation verknüpft sind. `AWSServiceRoleForAmazonInspector2Agentless`ist erforderlich, damit Amazon Inspector Vorgänge wie das Abrufen von Informationen über Ihre Amazon EC2 EC2-Instances und Amazon EBS-Snapshots ausführen kann. Es ist auch erforderlich, Amazon EBS-Snapshots zu entschlüsseln, die mit Schlüsseln verschlüsselt sind. AWS KMS `AWSServiceRoleForAmazonInspector2ThirdParty`ist erforderlich, damit Amazon Inspector Multi-Cloud-Scanvorgänge durchführen kann, z. B. die Verwaltung von Cloud-Connectors und Konfigurationsrekordern. Ermöglicht außerdem die Erstellung der `AWSServiceRoleForConfigThirdParty` serviceverknüpften Rolle, die für die Aufzeichnung von Multi-Cloud-Ressourcen erforderlich ist. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon Inspector.

- `config`— Ermöglicht Amazon Inspector, Konnektoren für die Multi-Cloud-Ressourcenerkennung zu erstellen, zu lesen und aufzulisten.
- `organizations`— Ermöglicht Administratoren die Verwendung von Amazon Inspector für eine Organisation in AWS Organizations. Wenn Sie den vertrauenswürdigen Zugriff für Amazon Inspector in aktivieren AWS Organizations, können Mitglieder des delegierten Administratorkontos Einstellungen verwalten und Ergebnisse in ihrer gesamten Organisation einsehen.
- `codeguru-security`— Ermöglicht Administratoren, Amazon Inspector zu verwenden, um Informationscodefragmente abzurufen und die Verschlüsselungseinstellungen für Code zu ändern, den CodeGuru Security speichert. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand für den Code in Ihren Ergebnissen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie AmazonInspector2FullAccessim Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonInspector2ReadOnlyAccess

Sie können die `AmazonInspector2ReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Berechtigungen, die nur Lesezugriff auf Amazon Inspector ermöglichen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `inspector2`— Ermöglicht den schreibgeschützten Zugriff auf die Funktionen von Amazon Inspector.
- `organizations`— Ermöglicht die Anzeige von Details zum Versicherungsschutz durch Amazon Inspector für ein Unternehmen. AWS Organizations Darüber hinaus können Sie die Unternehmensrichtlinien von Inspector anzeigen, indem Sie nach Inspector-Richtlinientypen filtern, die Ressourcenrichtlinien für die Delegierung anzeigen und die effektiven Inspector-Richtlinien anzeigen`DescribeResourcePolicy`, die auf Konten angewendet wurden`DescribeEffectivePolicy`. `ListPolicies` Auf diese Weise können Benutzer die zentralisierte Aktivierung von Inspector anhand von Unternehmensrichtlinien nachvollziehen, ohne diese ändern zu müssen.
- `codeguru-security`— Ermöglicht das Abrufen von Codefragmenten aus der Sicherheitsabteilung. CodeGuru Ermöglicht auch das Einsehen der Verschlüsselungseinstellungen für Ihren in CodeGuru Security gespeicherten Code.
- `config`— Ermöglicht die Anzeige der Details und des Gesundheitsstatus aller Amazon Inspector Multi-Cloud-Konnektoren. Dies beinhaltet die Auflistung und Beschreibung aller Konfigurationsrekorder, die den Konnektoren zugeordnet sind.
- `ssm`— Ermöglicht die Anzeige der Details und des Gesundheitsstatus aller Amazon Inspector Multi-Cloud-Konnektoren.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie AmazonInspector2ReadOnlyAccessim Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonInspector2ManagedCisPolicy

Sie können die `AmazonInspector2ManagedCisPolicy`-Richtlinie auch Ihren IAM-Entitäten anfügen. Diese Richtlinie sollte einer Rolle zugeordnet werden, die Ihren Amazon EC2 EC2-Instances die Erlaubnis erteilt, CIS-Scans der Instance auszuführen. Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie all ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `inspector2`— Ermöglicht den Zugriff auf Aktionen, die zur Ausführung von CIS-Scans verwendet werden.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie AmazonInspector2ManagedCisPolicyim Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonInspector2ServiceRolePolicy

Sie können die `AmazonInspector2ServiceRolePolicy`-Richtlinie Ihren IAM-Entitäten nicht anfügen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es Amazon Inspector ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon Inspector.

AWS verwaltete Richtlinie: AmazonInspector2AgentlessServiceRolePolicy

Sie können die `AmazonInspector2AgentlessServiceRolePolicy`-Richtlinie Ihren IAM-Entitäten nicht anfügen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es Amazon Inspector ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon Inspector.

AWS verwaltete Richtlinie: AmazonInspector2ThirdPartyServiceRolePolicy

Sie können die `AmazonInspector2ThirdPartyServiceRolePolicy`-Richtlinie Ihren IAM-Entitäten nicht anfügen. Diese Richtlinie ist mit einer servicebezogenen Rolle verknüpft, die es Amazon Inspector ermöglicht, in Ihrem Namen Aktionen für das Scannen von Multi-Cloud-Ressourcen durchzuführen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon Inspector.

AWS verwaltete Richtlinie: AmazonInspector2ManagedTelemetryPolicy

Sie können die `AmazonInspector2ManagedTelemetryPolicy`-Richtlinie auch Ihren IAM-Entitäten anfügen. Diese Richtlinie gewährt Berechtigungen für Telemetrieoperationen von Amazon Inspector, sodass der Service Paketinventardaten für Sicherheitslücken sammeln und übertragen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `inspector2-telemetry`— Ermöglicht den Zugriff auf Aktionen zur Übertragung von Paketinventardaten.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AmazonInspector2ManagedTelemetryPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

Amazon Inspector aktualisiert auf AWS Verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon Inspector an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Amazon Inspector Document History-Seite.

Änderungen	Beschreibung	Date
AmazonInspector2FullAccess_v2 — Aktualisierungen einer bestehenden Richtlinie	Amazon Inspector hat `iam` Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, die serviceverknüpfte Rolle für Multi-Cloud-Connectors (`AWSServiceRoleForAmazonInspector2ThirdParty`) und die serviceverknüpfte Drittanbieter-Rolle () zu erstellen. `AWSServiceRoleForConfigThirdParty` Amazon Inspector hat außerdem den Multi-Cloud-Service Principal (`multicloud.inspector2.amazonaws.com`)	7. Juli 2026

Änderungen	Beschreibung	Date
	zu den Bedingungen für `organizations` Aktionen hinzugefügt und Connector-Berechtigungen für die Multi-Cloud-Ressourcenerkennung hinzugefügt.	
AmazonInspector2ReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat Schreibschutz und AWS Systems Manager Berechtigungen für die Anzeige des Integritätsstatus von Multi-Cloud-Connectoren hinzugefügt.	7. Juli 2026
AWSInspector2OrganizationsAccess – Neue Richtlinie	Amazon Inspector hat eine neue verwaltete Richtlinie hinzugefügt, die die für die Aktivierung und Verwaltung von Amazon Inspector erforderlichen Berechtigungen per AWS Organizations Richtlinie gewährt.	3. März 2026
AmazonInspector2ManagedTelemetryPolicy – Neue Richtlinie	Amazon Inspector hat eine neue verwaltete Richtlinie hinzugefügt, die Berechtigungen für Amazon Inspector-Telemetrieoperationen gewährt, sodass der Service Paketinventardaten für Sicherheitslücken sammeln und übertragen kann.	5. Februar 2026

Änderungen	Beschreibung	Date
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat eine neue Berechtigung hinzugefügt, die es Amazon Inspector ermöglicht, Firewall-Metadaten für die Analyse der Netzwerkerreichbarkeit zu beschreiben. Darüber hinaus hat Amazon Inspector zusätzlichen Ressourcenbereich hinzugefügt, damit Amazon Inspector SSM-Verknüpfungen mit SSM-Dokumenten erstellen, aktualisieren und starten kann. `AWS-ConfigureAWSPackage`	3. Februar 2026
AmazonInspector2FullAccess_v2 und AmazonInspector2ReadOnlyAccess— Aktualisierungen vorhandener Richtlinien	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es den Versicherungsnehmern ermöglichen, die Unternehmensrichtlinien und Delegierungskonfigurationen von Inspector einzusehen. Dies unterstützt die zentrale Verwaltung und Transparenz der Inspector-Aktivierung durch AWS Organizations Richtlinien.	14. November 2025

Änderungen	Beschreibung	Date
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es der Amazon AWS Organizations Inspector-Richtlinie ermöglichen, die Aktivierung und Deaktivierung von Amazon Inspector durchzusetzen.	10. November 2025
AmazonInspector2FullAccess_v2 — Neue Richtlinie	Amazon Inspector hat eine neue verwaltete Richtlinie hinzugefügt, die vollen Zugriff auf Amazon Inspector und Zugriff auf andere verwandte Dienste bietet.	03. Juli 2025
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat eine neue Berechtigung hinzugefügt, die es Amazon Inspector ermöglicht, IP-Adressen und Internet-Gateways zu beschreiben.	29. April 2025
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die nur Lesezugriff auf Amazon ECS- und Amazon EKS-Aktionen ermöglichen.	25. März 2025
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, Funktions-Tags zurückzugeben AWS Lambda.	31. Juli 2024

Änderungen	Beschreibung	Date
AmazonInspector2FullAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, die serviceverknüpfte Rolle zu erstellen . `AWSServiceRoleForAmazonInspector2Agentless` Dadurch können Benutzer agentenbasiertes Scannen und agentenloses Scannen durchführen, wenn sie Amazon Inspector aktivieren.	24. April 2024
AmazonInspector2ManagedCisPolicy – Neue Richtlinie	Amazon Inspector hat eine neue verwaltete Richtlinie hinzugefügt, die Sie als Teil eines Instance-Profils verwenden können, um CIS-Scans auf einer Instance zuzulassen.	23. Januar 2024
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, CIS-Scans auf Ziel-Instances zu starten.	23. Januar 2024
AmazonInspector2AgentlessServiceRolePolicy – Neue Richtlinie	Amazon Inspector hat eine neue servicebezogene Rollenrichtlinie hinzugefügt, um das agentenlose Scannen von EC2-Instances zu ermöglichen.	27. November 2023

Änderungen	Beschreibung	Date
AmazonInspector2ReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Benutzern mit Lesezugriff ermöglichen, Informationen zu Sicherheitslücken für gefundene Sicherheitslücken in Paketen abzurufen.	22. September 2023
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, Netzwerkkonfigurationen von Amazon EC2 EC2-Instances zu scannen, die Teil der Elastic Load Balancing Balancing-Zielgruppen sind.	31. August 2023
AmazonInspector2ReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Benutzern mit Lesezugriff ermöglichen, Software Bill of Materials (SBOM) für ihre Ressourcen zu exportieren.	29. Juni 2023
AmazonInspector2ReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Benutzern mit Lesezugriff ermöglichen, Details der Verschlüsselungseinstellungen für Lambda-Code-Scanergebnisse für ihr Konto abzurufen.	13. Juni 2023

Änderungen	Beschreibung	Date
AmazonInspector2FullAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, mit denen Benutzer einen vom Kunden verwalteten KMS-Schlüssel konfigurieren können, um Code in Ergebnissen aus Lambda-Code-Scans zu verschlüsseln.	13. Juni 2023
AmazonInspector2ReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Benutzern mit Lesezugriff ermöglichen, Details zum Status und zu den Ergebnissen des Lambda-Code-Scans für ihr Konto abzurufen.	02. Mai 2023
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, AWS CloudTrail serviceverknüpfte Kanäle in Ihrem Konto zu erstellen, wenn Sie Lambda-Scanning aktivieren. Dadurch kann Amazon Inspector CloudTrail Ereignisse in Ihrem Konto überwachen.	30. April 2023

Änderungen	Beschreibung	Date
AmazonInspector2FullAccess – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Benutzern ermöglichen, Details zu den beim Lambda-Code-Scannen gefundenen Sicherheitslücken abzurufen.	21. April 2023
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, Informationen über die benutzerdefinierten Pfade, die ein Kunde für Amazon EC2 Deep Inspection definiert hat, an Amazon EC2 Systems Manager zu senden.	17. April 2023
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, AWS CloudTrail serviceverknüpfte Kanäle in Ihrem Konto zu erstellen, wenn Sie Lambda-Scanning aktivieren. Dadurch kann Amazon Inspector CloudTrail Ereignisse in Ihrem Konto überwachen.	30. April 2023

Änderungen	Beschreibung	Date
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, Scans des Entwicklercodes in AWS Lambda Funktionen anzufordern und Scandaten von Amazon CodeGuru Security zu empfangen. Darüber hinaus hat Amazon Inspector Berechtigungen zur Überprüfung von IAM-Richtlinien hinzugefügt. Amazon Inspector verwendet diese Informationen, um Lambda-Funktionen auf Code-Schwachstellen zu überprüfen.	28. Februar 2023
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat eine neue Anweisung hinzugefügt, die es Amazon Inspector ermöglicht, Informationen CloudWatch darüber abzurufen, wann eine AWS Lambda Funktion zuletzt aufgerufen wurde. Amazon Inspector verwendet diese Informationen, um Scans auf die Lambda-Funktionen in Ihrer Umgebung zu konzentrieren, die in den letzten 90 Tagen aktiv waren.	20. Februar 2023

Änderungen	Beschreibung	Date
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat eine neue Anweisung hinzugefügt, die es Amazon Inspector ermöglicht, Informationen über AWS Lambda Funktionen abzurufen, einschließlich jeder Layer-Version, die jeder Funktion zugeordnet ist. Amazon Inspector verwendet diese Informationen, um Lambda-Funktionen auf Sicherheitslücken zu überprüfen.	28. November 2022
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat eine neue Aktion hinzugefügt, mit der Amazon Inspector die Ausführung von SSM-Verknüpfungen beschreiben kann. Darüber hinaus hat Amazon Inspector zusätzlichen Ressourcenbereich hinzugefügt, damit Amazon Inspector SSM-Verknüpfungen mit `AmazonInspector2` eigenen SSM-Dokumenten erstellen, aktualisieren, löschen und starten kann.	31. August 2022
AmazonInspector2ServiceRolePolicyAktualisierungen einer bestehenden Richtlinie	Amazon Inspector hat den Ressourcenbereich der Richtlinie aktualisiert, sodass Amazon Inspector Softwareinventar in anderen AWS Partitionen erfassen kann.	12. August 2022

Änderungen	Beschreibung	Date
AmazonInspector2ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon Inspector hat den Ressourcenbereich der Aktionen neu strukturiert, sodass Amazon Inspector SSM-Verknüpfungen erstellen, löschen und aktualisieren kann.	10. August 2022
AmazonInspector2ReadOnlyAccess – Neue Richtlinie	Amazon Inspector hat eine neue Richtlinie hinzugefügt, die den schreibgeschützten Zugriff auf die Funktionen von Amazon Inspector ermöglicht.	21. Januar 2022
AmazonInspector2FullAccess – Neue Richtlinie	Amazon Inspector hat eine neue Richtlinie hinzugefügt, um vollen Zugriff auf die Funktionen von Amazon Inspector zu ermöglichen.	29. November 2021
AmazonInspector2ServiceRolePolicy – Neue Richtlinie	Amazon Inspector hat eine neue Richtlinie hinzugefügt, die es Amazon Inspector ermöglicht, in Ihrem Namen Aktionen in anderen Diensten durchzuführen.	29. November 2021
Amazon Inspector hat begonnen, Änderungen nachzuverfolgen	Amazon Inspector hat damit begonnen, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	29. November 2021

Verwenden von serviceverknüpften Rollen für Amazon Inspector

Amazon Inspector verwendet eine AWS Identity and Access Management (IAM) -Serviceverknüpfte Rolle mit dem Namen. `AWSServiceRoleForAmazonInspector2` Bei dieser serviceverknüpften Rolle handelt es sich um eine IAM-Rolle, die direkt mit Amazon Inspector verknüpft ist. Es ist von Amazon Inspector vordefiniert und beinhaltet alle Berechtigungen, die Amazon Inspector benötigt, um andere in AWS-Services Ihrem Namen anzurufen.

Eine serviceverknüpfte Rolle erleichtert die Einrichtung von Amazon Inspector, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon Inspector definiert die Berechtigungen seiner serviceverknüpften Rolle. Sofern nicht anders definiert, kann nur Amazon Inspector die Rolle übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. eine Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Service-linked Rollenberechtigungen im IAM-Benutzerhandbuch. Sie können eine mit einem Dienst verknüpfte Rolle erst löschen, nachdem Sie die zugehörigen Ressourcen gelöscht haben. Dadurch werden Ihre Amazon Inspector Inspector-Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Diensten, die dienstbezogene Rollen unterstützen, finden Sie unter AWS Dienste, die mit IAM funktionieren. Suchen Sie in der Rollenspalte nach den Diensten, für die Ja steht. Service-linked Wählen Sie Ja mit einem Link aus, um die Dokumentation zu serviceverknüpften Rollen für diesen Dienst zu lesen.

Service-linked Rollenberechtigungen für Amazon Inspector

Amazon Inspector verwendet die verwaltete Richtlinie mit dem Namen `AWSServiceRoleForAmazonInspector2`. Diese dienstbezogene Rolle vertraut darauf, dass der `inspector2.amazonaws.com` Service die Rolle übernimmt.

Die Berechtigungsrichtlinie für die Rolle, die benannt ist `AmazonInspector2ServiceRolePolicy`, ermöglicht es Amazon Inspector, Aufgaben wie die folgenden auszuführen:

- Verwenden Sie Amazon Elastic Compute Cloud (Amazon EC2) -Aktionen, um Informationen über Ihre Instances und Netzwerkpfade abzurufen.

- Verwenden Sie AWS Systems Manager Aktionen, um Inventar von Ihren Amazon EC2 EC2-Instances abzurufen und Informationen über Pakete von Drittanbietern aus benutzerdefinierten Pfaden abzurufen.
- Verwenden Sie die AWS Systems Manager `SendCommand` Aktion, um CIS-Scans für Ziel-Instances aufzurufen.
- Verwenden Sie Amazon Elastic Container Registry-Aktionen, um Informationen über Ihre Container-Images abzurufen.
- Verwenden Sie AWS Lambda Aktionen, um Informationen über Ihre Lambda-Funktionen abzurufen.
- Verwenden Sie AWS Organizations Aktionen, um zugehörige Konten zu beschreiben.
- Verwenden Sie CloudWatch Aktionen, um Informationen darüber abzurufen, wann Ihre Lambda-Funktionen zuletzt aufgerufen wurden.
- Verwenden Sie ausgewählte IAM-Aktionen, um Informationen über Ihre IAM-Richtlinien abzurufen, die zu Sicherheitslücken in Ihrem Lambda-Code führen könnten.
- Verwenden Sie Amazon Q-Aktionen, um den Code in Ihren Lambda-Funktionen zu scannen. Amazon Inspector verwendet die folgenden Amazon Q-Aktionen:
  - codeguru-security: CreateScan — Erteilt die Erlaubnis, einen Amazon Q; -Scan zu erstellen.
  - codeguru-security: GetScan — Erteilt die Erlaubnis, Amazon Q-Scan-Metadaten abzurufen.
  - codeguru-security: ListFindings — Erteilt die Erlaubnis, von Amazon Q generierte Ergebnisse abzurufen.
  - codeguru-security: DeleteScansByCategory — Erteilt Amazon Q die Erlaubnis, von Amazon Inspector initiierte Scans zu löschen.
  - codeguru-security: BatchGetFindings — Erteilt die Erlaubnis, eine Reihe von spezifischen Ergebnissen abzurufen, die von Amazon Q generiert wurden.
- Verwenden Sie ausgewählte Elastic Load Balancing Balancing-Aktionen, um Netzwerkscans von EC2-Instances durchzuführen, die Teil der Elastic Load Balancing Balancing-Zielgruppen sind.
- Verwenden Sie Amazon ECS- und Amazon EKS-Aktionen, um nur Lesezugriff zu gewähren, um Cluster und Aufgaben anzuzeigen und Aufgaben zu beschreiben.
- Verwenden Sie AWS Organizations Aktionen, um delegierte Administratoren für Amazon Inspector organisationsübergreifend aufzulisten.
- Verwenden Sie Amazon Inspector-Aktionen, um Amazon Inspector organisationsübergreifend zu aktivieren und zu deaktivieren.

- Verwenden Sie Amazon Inspector Inspector-Aktionen, um delegierte Administratorkonten zu benennen und Mitgliedskonten organisationsübergreifend zuzuordnen.

> **Note**
>
> Amazon Inspector verwendet CodeGuru keine Lambda-Scans mehr. AWS wird den Support für den 20. CodeGuru November 2025 einstellen. Weitere Informationen finden Sie unter Ende des Supports für CodeGuru Security. Amazon Inspector verwendet jetzt Amazon Q zur Durchführung von Lambda-Scans und benötigt nicht die in diesem Abschnitt beschriebenen Berechtigungen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie AmazonInspector2ServiceRolePolicyim Referenzhandbuch für AWS verwaltete Richtlinien.

Eine serviceverknüpfte Rolle für Amazon Inspector erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie Amazon Inspector in der AWS-Managementkonsole, der oder der AWS API aktivieren AWS CLI, erstellt Amazon Inspector die serviceverknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für Amazon Inspector

Amazon Inspector erlaubt Ihnen nicht, die `AWSServiceRoleForAmazonInspector2` serviceverknüpfte Rolle zu bearbeiten. Nachdem eine serviceverknüpfte Rolle erstellt wurde, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten möglicherweise auf die Rolle verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon Inspector

Wenn Sie Amazon Inspector nicht mehr verwenden müssen, empfehlen wir Ihnen, die `AWSServiceRoleForAmazonInspector2` serviceverknüpfte Rolle zu löschen. Bevor Sie die Rolle löschen können, müssen Sie Amazon Inspector in allen Bereichen deaktivieren, in AWS-Region denen sie aktiviert ist. Wenn Sie Amazon Inspector deaktivieren, wird die Rolle nicht für Sie gelöscht. Wenn Sie Amazon Inspector erneut aktivieren, kann Amazon Inspector daher die bestehende Rolle verwenden. Auf diese Weise können Sie vermeiden, dass eine ungenutzte Entität nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Wenn Sie diese serviceverknüpfte Rolle löschen und dann erneut erstellen müssen, können Sie die Rolle in Ihrem Konto mit demselben Verfahren neu anlegen. Wenn Sie Amazon Inspector aktivieren, erstellt Amazon Inspector die serviceverknüpfte Rolle für Sie neu.

> **Note**
>
> Wenn der Amazon Inspector-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Warten Sie in diesem Fall einige Minuten und führen Sie den Vorgang dann erneut aus.

Sie können die IAM-Konsole, die oder die AWS API verwenden AWS CLI, um die `AWSServiceRoleForAmazonInspector2` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Service-linked Rollenberechtigungen für agentenlose Amazon Inspector-Scans

Das agentenlose Scannen von Amazon Inspector verwendet die angegebene, mit dem Service verknüpfte Rolle. `AWSServiceRoleForAmazonInspector2Agentless` Mit dieser Spiegelreflexkamera kann Amazon Inspector einen Amazon EBS-Volume-Snapshot in Ihrem Konto erstellen und dann auf die Daten aus diesem Snapshot zugreifen. Diese dienstbezogene Rolle vertraut darauf, dass der `agentless.inspector2.amazonaws.com` Service die Rolle übernimmt.

> **Important**
>
> Die Anweisungen in dieser servicebezogenen Rolle verhindern, dass Amazon Inspector agentenlose Scans auf allen EC2-Instances durchführt, die Sie mithilfe des Tags von Scans ausgeschlossen haben. `InspectorEc2Exclusion` Darüber hinaus verhindern die Anweisungen, dass Amazon Inspector auf verschlüsselte Daten von einem Volume zugreift, wenn der für die Verschlüsselung verwendete KMS-Schlüssel das `InspectorEc2Exclusion` Tag trägt. Weitere Informationen finden Sie unter Instances von Amazon Inspector-Scans ausschließen.

Die Berechtigungsrichtlinie für die Rolle, die benannt ist`AmazonInspector2AgentlessServiceRolePolicy`, ermöglicht es Amazon Inspector, Aufgaben wie die folgenden auszuführen:

- Verwenden Sie Amazon Elastic Compute Cloud (Amazon EC2) -Aktionen, um Informationen über Ihre EC2-Instances, Volumes und Snapshots abzurufen.
  - Verwenden Sie Amazon EC2-Tagging-Aktionen, um Schnappschüsse für Scans mit dem `InspectorScan` Tag-Schlüssel zu taggen.
  - Verwenden Sie Amazon EC2-Snapshot-Aktionen, um Snapshots zu erstellen, sie mit dem `InspectorScan` Tag-Schlüssel zu kennzeichnen und anschließend Snapshots von Amazon EBS-Volumes zu löschen, die mit dem Tag-Schlüssel gekennzeichnet wurden. `InspectorScan`
- Verwenden Sie Amazon EBS-Aktionen, um Informationen aus Snapshots abzurufen, die mit dem `InspectorScan` Tag-Schlüssel gekennzeichnet sind.
- Verwenden Sie ausgewählte AWS KMS Entschlüsselungsaktionen, um Snapshots zu entschlüsseln, die mit vom Kunden verwalteten Schlüsseln verschlüsselt wurden. AWS KMS Amazon Inspector entschlüsselt keine Snapshots, wenn der KMS-Schlüssel, mit dem sie verschlüsselt wurden, mit dem Tag gekennzeichnet ist. `InspectorEc2Exclusion`

Die Rolle ist mit der folgenden Berechtigungsrichtlinie konfiguriert.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "InstanceIdentification",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeVolumes",
    "ec2:DescribeSnapshots"
   ],
   "Resource": "*"
  },
  {
   "Sid": "GetSnapshotData",
   "Effect": "Allow",
   "Action": [
    "ebs:ListSnapshotBlocks",
    "ebs:GetSnapshotBlock"
   ],
```

```
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "StringLike": {
      "aws:ResourceTag/InspectorScan": "*"
     }
    }
   },
   {
    "Sid": "CreateSnapshotsAnyInstanceOrVolume",
    "Effect": "Allow",
    "Action": "ec2:CreateSnapshots",
    "Resource": [
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*"
    ]
   },
   {
    "Sid": "DenyCreateSnapshotsOnExcludedInstances",
    "Effect": "Deny",
    "Action": "ec2:CreateSnapshots",
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
     "StringEquals": {
      "ec2:ResourceTag/InspectorEc2Exclusion": "true"
     }
    }
   },
   {
    "Sid": "CreateSnapshotsOnAnySnapshotOnlyWithTag",
    "Effect": "Allow",
    "Action": "ec2:CreateSnapshots",
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "Null": {
      "aws:TagKeys": "false"
     },
     "ForAllValues:StringEquals": {
      "aws:TagKeys": "InspectorScan"
     }
    }
   },
   {
    "Sid": "CreateOnlyInspectorScanTagOnlyUsingCreateSnapshots",
    "Effect": "Allow",
```

```
   "Action": "ec2:CreateTags",
   "Resource": "arn:aws:ec2:*:*:snapshot/*",
   "Condition": {
    "StringLike": {
     "ec2:CreateAction": "CreateSnapshots"
    },
    "Null": {
     "aws:TagKeys": "false"
    },
    "ForAllValues:StringEquals": {
     "aws:TagKeys": "InspectorScan"
    }
   }
  },
  {
   "Sid": "DeleteOnlySnapshotsTaggedForScanning",
   "Effect": "Allow",
   "Action": "ec2:DeleteSnapshot",
   "Resource": "arn:aws:ec2:*:*:snapshot/*",
   "Condition": {
    "StringLike": {
     "ec2:ResourceTag/InspectorScan": "*"
    }
   }
  },
  {
   "Sid": "DenyKmsDecryptForExcludedKeys",
   "Effect": "Deny",
   "Action": "kms:Decrypt",
   "Resource": "arn:aws:kms:*:*:key/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/InspectorEc2Exclusion": "true"
    }
   }
  },
  {
   "Sid": "DecryptSnapshotBlocksVolContext",
   "Effect": "Allow",
   "Action": "kms:Decrypt",
   "Resource": "arn:aws:kms:*:*:key/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
    },
    "StringLike": {
     "kms:ViaService": "ec2.*.amazonaws.com",
     "kms:EncryptionContext:aws:ebs:id": "vol-*"
    }
   }
  },
  {
   "Sid": "DecryptSnapshotBlocksSnapContext",
   "Effect": "Allow",
   "Action": "kms:Decrypt",
   "Resource": "arn:aws:kms:*:*:key/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    },
    "StringLike": {
     "kms:ViaService": "ec2.*.amazonaws.com",
     "kms:EncryptionContext:aws:ebs:id": "snap-*"
    }
   }
  },
  {
   "Sid": "DescribeKeysForEbsOperations",
   "Effect": "Allow",
   "Action": "kms:DescribeKey",
   "Resource": "arn:aws:kms:*:*:key/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    },
    "StringLike": {
     "kms:ViaService": "ec2.*.amazonaws.com"
    }
   }
  },
  {
   "Sid": "ListKeyResourceTags",
   "Effect": "Allow",
   "Action": "kms:ListResourceTags",
   "Resource": "arn:aws:kms:*:*:key/*"
  }
 ]
}
```

Erstellung einer dienstbezogenen Rolle für agentenloses Scannen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie Amazon Inspector in der AWS-Managementkonsole, der oder der AWS API aktivieren AWS CLI, erstellt Amazon Inspector die serviceverknüpfte Rolle für Sie.

Bearbeitung einer serviceverknüpften Rolle für agentenloses Scannen

Amazon Inspector erlaubt Ihnen nicht, die `AWSServiceRoleForAmazonInspector2Agentless` serviceverknüpfte Rolle zu bearbeiten. Nachdem eine serviceverknüpfte Rolle erstellt wurde, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten möglicherweise auf die Rolle verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für das Scannen ohne Agenten

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird.

> **Important**
>
> Um die `AWSServiceRoleForAmazonInspector2Agentless` Rolle zu löschen, müssen Sie Ihren Scanmodus in allen Regionen, in denen agentenloses Scannen verfügbar ist, auf agentenbasiert einstellen.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die serviceverknüpfte Rolle zu löschen. AWSServiceRoleForAmazonInspector2Agentless Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Service-linked Rollenberechtigungen für Amazon Inspector-Multi-Cloud-Scans

Das Multi-Cloud-Scannen von Amazon Inspector verwendet die angegebene serviceverknüpfte Rolle. `AWSServiceRoleForAmazonInspector2ThirdParty` Mit dieser Spiegelreflexkamera

kann Amazon Inspector in Ihrem Namen Aktionen für das Scannen von Multi-Cloud-Ressourcen durchführen, einschließlich der Verwaltung von Cloud-Connectoren, Konfigurationsrekordern und Schwachstellenanalysen von Ressourcen in externen Cloud-Umgebungen. Diese dienstbezogene Rolle vertraut darauf, dass der `thirdparty.inspector2.amazonaws.com` Service die Rolle übernimmt.

Die Berechtigungsrichtlinie für die Rolle, die benannt ist`AmazonInspector2ThirdPartyServiceRolePolicy`, ermöglicht es Amazon Inspector, Aufgaben wie die folgenden auszuführen:

- Verwenden Sie Aktionen, um Multi-Cloud-Ressourcen mithilfe von Inspector-owned Serviceansichten zu entdecken.
- Verwenden Sie AWS Systems Manager Aktionen, um Verknüpfungen, Automatisierungsausführungen und Cloud-Connectors für Multi-Cloud-VM-Scans zu verwalten.
- Verwenden Sie die `PassRole` IAM-Aktion, um Amazon Inspector-Rollen AWS Systems Manager zur automatisierten Ausführung zu übergeben, die auf Rollen beschränkt sind, deren Namen mit oder beginnen. `Inspector2SSM` `Inspector2VmScanner`
- Verwenden Sie die `GetRole` IAM-Aktion, um zu überprüfen, ob Amazon Inspector-Rollen existieren, bevor Sie sie weitergeben.
- Verwenden Sie die `CreateServiceLinkedRole` IAM-Aktion, um die AWS Systems Manager serviceverknüpfte Rolle zu erstellen, falls dies für die Erstellung der Zuordnung erforderlich ist.
- Verwenden Sie Aktionen, um serviceverknüpfte Konfigurationsrekorder und Konnektoren für die Erkennung von Multi-Cloud-Ressourcen zu verwalten.
- Verwenden Sie CloudWatch Aktionen, um Integritätskennzahlen für Multi-Cloud-Connectors abzurufen.
- Verwenden Sie AWS STS `GetWebIdentityToken` Action für den OIDC-Verbund mit externen Cloud-Umgebungen, die auf die Azure AD-Token-Zielgruppe zugeschnitten sind.

Informationen zu den in dieser Richtlinie enthaltenen Berechtigungen finden Sie AmazonInspector2ThirdPartyServiceRolePolicyim Referenzhandbuch für AWS verwaltete Richtlinien.

Eine serviceverknüpfte Rolle für Multi-Cloud-Scanning erstellen

Sie müssen die serviceverknüpfte Rolle `AWSServiceRoleForAmazonInspector2ThirdParty` nicht manuell erstellen. Wenn Sie in Amazon Inspector einen Multi-Cloud-Connector erstellen, erstellt Amazon Inspector automatisch die serviceverknüpfte Rolle für Sie.

Bearbeitung einer serviceverknüpften Rolle für Multi-Cloud-Scanning

Amazon Inspector erlaubt Ihnen nicht, die `AWSServiceRoleForAmazonInspector2ThirdParty` serviceverknüpfte Rolle zu bearbeiten. Nachdem eine serviceverknüpfte Rolle erstellt wurde, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten möglicherweise auf die Rolle verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer dienstbezogenen Rolle für Multi-Cloud-Scanning

Wenn Sie das Multi-Cloud-Scannen nicht mehr verwenden müssen, empfehlen wir Ihnen, die dienstverknüpfte Rolle zu löschen. `AWSServiceRoleForAmazonInspector2ThirdParty` Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird.

> **Important**
>
> Um die `AWSServiceRoleForAmazonInspector2ThirdParty` Rolle zu löschen, müssen Sie zunächst alle Multi-Cloud-Connectors in allen Regionen löschen, in denen Multi-Cloud-Scanning aktiviert ist.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die serviceverknüpfte Rolle zu löschen. AWSServiceRoleForAmazonInspector2ThirdParty Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Fehlerbehebung bei Identität und Zugriff auf Amazon Inspector

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Inspector und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion in Amazon Inspector durchzuführen
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb meiner Umgebung zulassen AWS-Konto um auf meine Amazon Inspector Inspector-Ressourcen zuzugreifen

Ich bin nicht berechtigt, eine Aktion in Amazon Inspector durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `inspector2:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 inspector2:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `inspector2:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon Inspector übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Inspector auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meiner Umgebung zulassen AWS-Konto um auf meine Amazon Inspector Inspector-Ressourcen zuzugreifen

Sie können eine Rolle erstellen, mit der Benutzer anderer Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon Inspector diese Funktionen unterstützt, finden Sie unterSo arbeitet Amazon Inspector mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Überwachung von Amazon Inspector

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Verfügbarkeit, Zuverlässigkeit und Leistung von Amazon Inspector und anderen AWS Lösungen. AWS bietet Tools zur Überwachung von Amazon Inspector, zur Meldung auftretender Probleme und zur Ergreifung von Maßnahmen zur Behebung dieser Probleme:

- Amazon EventBridge ist ein AWS Service, der Ereignisse verwendet, um Anwendungskomponenten miteinander zu verbinden, sodass Sie leichter skalierbare, ereignisgesteuerte Anwendungen erstellen können. EventBridge stellt einen Stream von Echtzeitdaten aus Ihren Anwendungen, Software-as-a-Service (SaaS) -Anwendungen sowie

AWS Diensten und Routen bereit, sodass Sie Ereignisse in Diensten überwachen und ereignisgesteuerte Architekturen erstellen können.

- AWS CloudTrailist ein AWS Dienst, der API-Aufrufe und zugehörige Ereignisse erfasst, die von Ihnen oder in Ihrem Namen getätigt wurden. AWS-Konto CloudTrail übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket, sodass Sie feststellen können, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus Anrufe getätigt wurden und wann die Anrufe stattfanden.

Protokollieren Amazon Inspector Inspector-API-Aufrufen mit AWS CloudTrail

Amazon Inspector ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem IAM-Benutzer oder einer IAM-Rolle oder einem AWS-Service in Amazon Inspector ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für Amazon Inspector als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon Inspector Inspector-Konsole und Aufrufe der Amazon Inspector Inspector-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon Inspector. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole in Event history (Ereignisverlauf) anzeigen. Anhand der von CloudTrail gesammelten Informationen können Sie Folgendes bestimmen:

- Die Anfrage, die an Amazon Inspector gestellt wurde.
- Die IP-Adresse, von der die Anforderung erfolgt ist.
- Wer die Anfrage gestellt hat.
- Wann die Anfrage gestellt wurde.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Informationen zu Amazon Inspector in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn in Amazon Inspector eine Aktivität auftritt, wird diese Aktivität zusammen mit anderen AWS-Service Ereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter Ereignisse mit dem CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich Ereignissen für Amazon Inspector, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS -Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere konfigurieren, AWS-Services um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie unter den folgenden Themen:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien von mehreren Konten
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen

Alle Amazon Inspector Inspector-Aktionen werden von protokolliert CloudTrail. Alle Aktionen, die Amazon Inspector ausführen kann, sind in der Amazon Inspector API-Referenz dokumentiert. Zum Beispiel werden durch Aufrufe der `CreateFindingsReport`-, `ListCoverage`- und `UpdateOrganizationConfiguration`-Aktionen Einträge in den CloudTrail -Protokolldateien generiert.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Stammbenutzers oder des IAM-Benutzers gestellt wurde.
- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer ausgeführt wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

Grundlegendes zu Amazon Inspector Inspector-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen

oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar. Zu den Ereignissen gehören Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Amazon Inspector Scaninformationen in CloudTrail

Amazon Inspector Scan ist in integriert CloudTrail. Alle Amazon Inspector Scan API-Operationen werden als Verwaltungsereignisse protokolliert. Eine Liste der Amazon Inspector Scan API-Operationen, bei denen Amazon Inspector protokolliert CloudTrail, finden Sie unter Amazon Inspector Scan in der Amazon Inspector API-Referenz.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `ScanSbom` Aktion demonstriert:


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA123456789EXAMPLE:akua_mansa",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/akua_mansa",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-10-17T15:22:59Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-10-17T16:02:34Z",
    "eventSource": "gamma-inspector-scan.amazonaws.com",
    "eventName": "ScanSbom",
```

```
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-sdk-java/2.20.162 Mac_OS_X/13.5.2 OpenJDK_64-
Bit_Server_VM/17.0.8+7-LTS Java/17.0.8 vendor/Amazon.com_Inc. io/sync http/
UrlConnection cfg/retry-mode/legacy",
    "requestParameters": {
        "sbom": {
            "specVersion": "1.5",
            "metadata": {
                "component": {
                    "name": "debian",
                    "type": "operating-system",
                    "version": "9"
                }
            },
            "components": [
                {
                    "name": "packageOne",
                    "purl": "pkg:deb/debian/packageOne@1.0.0?arch=x86_64&distro=9",
                    "type": "application"
                }
            ],
            "bomFormat": "CycloneDX"
        }
    },
    "responseElements": null,
    "requestID": "f041a27f-f33e-4f70-b09b-5fbc5927282a",
    "eventID": "abc8d1e4-d214-4f07-bc56-8a31be6e36fe",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Konformitätsvalidierung für Amazon Inspector

Unser neues AWS Anmeldeerlebnis ist nicht für regulierte Workloads konzipiert. Wenn Sie unser neues AWS Anmeldeformular nutzen, es aber AWS für regulierte Workloads nutzen möchten, können Sie sich für AWS (erweiterte) Funktionen registrieren oder erweiterte Funktionen für Ihre Umgebung aktivieren. AWS

Informationen darüber, ob AWS-Service ein in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter AWS-Services Umfang nach Compliance-Programm unter Umfang nach Compliance-Programm AWS-Services das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der AWS Sicherheitsdokumentation.

Resilienz in Amazon Inspector

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere, physisch getrennte und isolierte Availability Zones, die mit Netzwerken mit niedriger Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Infrastruktursicherheit in Amazon Inspector

Als verwalteter Service ist Amazon Inspector durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Inspector zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Reaktion auf Vorfälle in Amazon Inspector

Sicherheit hat bei AWS höchste Priorität. Wie im Modell der AWS gemeinsamen Verantwortung unter „Sicherheit der Cloud“ erwähnt, AWS ist er für den Schutz der Infrastruktur verantwortlich, auf der alle Dienste in der AWS Cloud ausgeführt werden. AWS ist auch für die Reaktion auf alle Vorfälle im Zusammenhang mit dem Amazon Inspector-Service verantwortlich.

Als AWS Kunde tragen Sie gemeinsam die Verantwortung für die Aufrechterhaltung der Sicherheit in der AWS Cloud. Das bedeutet, dass Sie die Sicherheit kontrollieren, die Sie implementieren möchten. Dazu gehören alle AWS Tools und Funktionen, auf die Sie zugreifen. Das bedeutet auch, dass Sie im Rahmen des Modells der gemeinsamen Verantwortung für die Reaktion auf Vorfälle verantwortlich sind.

Indem Sie eine Sicherheitsbasis einrichten, die alle Ziele für Ihre in der AWS Cloud ausgeführten Anwendungen erfüllt, können Sie Abweichungen erkennen, auf die Sie reagieren können. Da es sich bei der Reaktion auf Vorfälle um ein komplexes Thema handelt, sollten Sie sich die folgenden Ressourcen ansehen, um besser zu verstehen, welche Auswirkungen die Reaktion auf Vorfälle hat und wie sich Ihre Entscheidungen auf Ihre Unternehmensziele auswirken könnten: Leitfaden zur Reaktion auf AWSAWS Sicherheitsvorfälle, Best Practices im Bereich Sicherheit und AWS Cloud Adoption Framework: Security Perspective.

Sicherheitsforschung von Amazon Inspector

Amazon Inspector Security Research ist ein Programm zur kontinuierlichen Überwachung, das bösartige Pakete identifiziert, die in öffentlichen Paketregistern veröffentlicht wurden. Angriffe auf Software-Lieferketten, die auf Open-Source-Ökosysteme abzielen, stellen eine wachsende Bedrohung dar, und bösartige Pakete werden häufig erst dann aus vorgelagerten Registern entfernt, nachdem sie von ahnungslosen Entwicklern heruntergeladen wurden. Amazon Inspector Security Research trägt zum Schutz Ihrer Anwendungen bei, indem es diese Pakete erkennt, Warnhinweise veröffentlicht und diese Informationen in die Ergebnisse von Amazon Inspector einfließen lässt, sodass Workloads, die ein bekanntes bösartiges Paket verwenden, automatisch entdeckt werden.

Unterstützte öffentliche Paketregistrierungen:

- NPM — die Node.js Paketregistrierung (npmjs.com)
- PyPI — der Python-Paketindex (pypi.org)

Forschungsmethodik

Das Sicherheitsforschungsteam von Amazon Inspector kombiniert automatisierte Erkennungspipelines mit Expertenanalysen, um bösartige Pakete in allen unterstützten Registern zu identifizieren. Jedem bestätigten bösartigen Paket wird eine zugewiesen MAL-ID, als öffentliche Empfehlung dokumentiert und in die Ergebnisse von Amazon Inspector integriert, sodass Kunden benachrichtigt werden, wenn ihre Workloads betroffene Pakete verbrauchen. Es ist kein Eingreifen des Kunden erforderlich, um von dieser Untersuchung zu profitieren — Erkennungen werden automatisch durchgeführt.

Amazon Inspector ist ein beitragender Partner im Repository bösartiger Pakete der Open Source Security Foundation (OpenSSF). Von Amazon Inspector Security Research erstellte Empfehlungen werden in diesem offenen Datensatz veröffentlicht, sodass die breitere Open-Source-Community Zugriff auf dieselben Bedrohungsinformationen erhält, die Amazon Inspector zum Schutz AWS seiner Kunden verwendet. Die jedem Advisory MAL-ID zugewiesenen Advisory sind mit dem OpenSSF-Advisory-Format kompatibel.

Zusammenfassung der Entdeckung

In den folgenden Tabellen sind die Erkennungen bösartiger Pakete durch Amazon Inspector Security Research in allen unterstützten Registern zusammengefasst.

> **Note**
>
> Letzte Aktualisierung: 2026-05-13 21:00:00 UTC

Gesamtwerte der Lebenszeit nach Registern

Registrierung	Lifetime-Pakete identifiziert
NPM	188.538
PyPI	12
Gesamt	188.550

Letzte Aktivität von Registry

Zeitraum	Fenster (UTC)	NPM	PyPI	Gesamt
Diese Woche	2026-05-11.. 2026-05-17	61	12	73
Letzte Woche	2026-05-04.. 2026-05-10	84	0	84
Dieser Monat	2026-05	180	12	192
Letzter Monat	2026-04	619	0	619

Aktuelle Berichte über bösartige Pakete

In der folgenden Tabelle sind die zehn neuesten von Amazon Inspector Security Research veröffentlichten Hinweise zu bösartigen Paketen aufgeführt, sortiert nach Veröffentlichungsdatum.

Package name	MAL-ID	Registrierung	Datum der Erkennung
d4rktg	MAL-2026-3688	PyPI	2026-05-13
@dropout - ai/runtime	MAL-2026-3683	NPM	2026-05-13
amino.fix	MAL-2026-3686	PyPI	2026-05-13
@gusmano/reext	MAL-2026-3684	NPM	2026-05-12
wird ständig aktualisiert	MAL-2026-3685	PyPI	2026-05-12
@a91082900/test_package	MAL-2026-3680	NPM	2026-05-12
Kegelläufer	MAL-2026-3693	PyPI	2026-05-12
88 q	MAL-2026-3676	NPM	2026-05-12

Package name	MAL-ID	Registrierung	Datum der Erkennung
66o	MAL-2026-3674	NPM	2026-05-12
6 ccm	MAL-2026-3675	NPM	2026-05-12

Zugehörige Ressourcen

Die folgenden AWS Sicherheits-Blogbeiträge bieten zusätzlichen Kontext zu Amazon Inspector Security Research und aktuellen Kampagnen zur Bedrohung der Lieferkette:

- Amazon Inspector erkennt über 150.000 bösartige Pakete im Zusammenhang mit einer Token-Farming-Kampagne
- Schützt vor Angriffen auf die Lieferkette wie Chalk, Debug und Wurm Shai-Hulud
- Was AWS Security aus der Reaktion auf die jüngsten NPM-Bedrohungskampagnen in der Lieferkette gelernt hat

Greifen Sie über einen Schnittstellenendpunkt auf Amazon Inspector zu (AWS PrivateLink)

Sie können AWS PrivateLink es verwenden, um eine private Verbindung zwischen Ihrer VPC und Amazon Inspector herzustellen. Sie können auf Amazon Inspector zugreifen, als wäre es in Ihrer VPC, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder Direct Connect eine Verbindung zu verwenden. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um auf Amazon Inspector zuzugreifen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Dabei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Einstiegspunkt für den Datenverkehr dienen, der für Amazon Inspector bestimmt ist.

Weitere Informationen finden Sie AWS PrivateLink im Handbuch unter Access AWS-Services through.AWS PrivateLink

Überlegungen zu Amazon Inspector

Bevor Sie einen Schnittstellenendpunkt für Amazon Inspector einrichten, lesen Sie die Überlegungen im AWS PrivateLink Handbuch.

Amazon Inspector unterstützt Aufrufe all seiner API-Aktionen über den Schnittstellenendpunkt.

VPC-Endpunktrichtlinien werden für Amazon Inspector nicht unterstützt. Standardmäßig ist der vollständige Zugriff auf Amazon Inspector über den Schnittstellenendpunkt zulässig. Alternativ können Sie den Endpunkt-Netzwerkschnittstellen eine Sicherheitsgruppe zuordnen, um den Datenverkehr zu Amazon Inspector über den Schnittstellenendpunkt zu steuern.

Erstellen Sie einen Schnittstellenendpunkt für Amazon Inspector

Sie können einen Schnittstellenendpunkt für Amazon Inspector entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im AWS PrivateLink -Leitfaden.

Wenn Sie einen Schnittstellenendpunkt für Amazon Inspector erstellen, verwenden Sie einen der folgenden Servicenamen:

```
com.amazonaws.region.inspector2
```

```
com.amazonaws.region.inspector-scan
```

*region*Ersetzen Sie es durch den entsprechenden AWS-Region Code AWS-Region.

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an Amazon Inspector richten, indem Sie beispielsweise dessen standardmäßigen regionalen DNS-Namen verwenden, `service-name.us-east-1.amazonaws.com` oder `service-name.us-east-1.api.aws.com` für die USA Ost (Nord-Virginia).

Amazon Inspector Inspector-Integrationen

Amazon Inspector lässt sich in andere AWS Dienste integrieren. Diese Dienste können Daten von Amazon Inspector aufnehmen, sodass Sie Ihre Ergebnisse auf unterschiedliche Weise einsehen können. Sehen Sie sich die folgenden Integrationsoptionen an, um mehr zu erfahren.

Verwenden von Amazon Inspector mit AWS Organizations

AWS Organizationshilft Ihnen dabei, Ihre AWS Umgebung zentral zu verwalten und zu steuern. Sie können AWS Organizations Richtlinien verwenden, um Amazon Inspector automatisch für mehrere Konten in Ihrer Organisation zu aktivieren und zu verwalten.

Die Organisationsrichtlinien von Amazon Inspector ermöglichen Ihnen:

- Aktivieren Sie Amazon Inspector-Scantypen (EC2, ECR, Lambda, Code Repository) zentral in Ihrem Unternehmen
- Wenden Sie die Amazon Inspector Inspector-Aktivierung automatisch auf neue Konten an, die der Organisation beitreten
- Sorgen Sie für eine einheitliche Scanabdeckung in allen Unternehmenseinheiten
- Verhindern Sie, dass Mitgliedskonten erforderliche Scans deaktivieren

Unternehmensrichtlinien steuern die Aktivierung von Ressourcentypen, während delegierte Administratoren die Kontrolle über die Einstellungen für die Scankonfiguration behalten. Informationen darüber, wie Unternehmensrichtlinien mit delegierten Administrator- und Mitgliedskontenberechtigungen interagieren, finden Sie unter. Verwaltung mehrerer Konten in Amazon Inspector mit AWS Organizations Detaillierte Anweisungen zur Erstellung von Amazon Inspector Inspector-Richtlinien finden Sie in der AWS Organizations Dokumentation zu den Amazon Inspector Inspector-Richtlinien.

Integration von Amazon Inspector mit Amazon ECR

Amazon Elastic Container Registry (Amazon ECR) ist eine AWS verwaltete Container-Image-Registry, die private Registrierungen unterstützt. Private Registrys von Amazon ECR hosten Container-Images in einer hochverfügbaren und skalierbaren Architektur. Sie können Amazon Inspector verwenden, um Container-Images, die sich in Ihrem Amazon ECR-Repository befinden, nach anfälligen Betriebssystempaketen und Programmiersprachenpaketen zu durchsuchen. Weitere

Informationen finden Sie unter Integration von Amazon Inspector mit Amazon Elastic Container Registry (Amazon ECR).

Amazon Inspector Inspector-Integration mit AWS Security Hub CSPM

AWS Security Hub CSPMbietet einen umfassenden Überblick über Ihren Sicherheitsstatus AWS und hilft Ihnen dabei, Ihre Umgebung anhand von Industriestandards und Best Practices zu überprüfen. Security Hub CSPM sammelt Sicherheitsdaten von AWS Konten, Diensten und unterstützten Produkten. Mit Security Hub CSPM können Sie die Ergebnisdaten von Amazon Inspector aufnehmen und einen zentralen Speicherort für Ergebnisse in all Ihren integrierten AWS Services und AWS Partner Network-Produkten einrichten. Weitere Informationen finden Sie unter Amazon Inspector Inspector-Integration mit AWS Security Hub CSPM.

Integration von Amazon Inspector mit Amazon Elastic Container Registry (Amazon ECR)

Amazon Elastic Container Registry ist eine vollständig verwaltete Container-Registry, die Docker- und OCI-Images und AWS -Artefakte unterstützt. Wenn Sie Amazon ECR verwenden, können Sie Enhanced Scanning für Ihr Container-Register aktivieren. Wenn Sie das erweiterte Scannen aktivieren, erkennt Amazon Inspector Ihre Container-Images automatisch und scannt sie nach anfälligen Betriebssystemen und Programmiersprachenpaketen. Diese Integration ermöglicht es Ihnen, die Ergebnisse von Amazon Inspector für Container-Images einzusehen und die Häufigkeit und den Umfang der Scans in der Amazon ECR-Konsole zu verwalten. Weitere Informationen finden Sie unter Amazon ECR-Container-Images mit Amazon Inspector scannen.

Aktivierung der Integration

Sie können die Integration aktivieren, indem Sie das Amazon Inspector-Scannen über die Amazon Inspector-Konsole oder API aktivieren oder indem Sie Ihr Repository so konfigurieren, dass es Enhanced Scanning mit Amazon Inspector über die Amazon ECR-Konsole oder API verwendet.

Weitere Informationen zur Aktivierung der Integration über Amazon Inspector finden Sie unterAutomatisierte Scantypen in Amazon Inspector.

Informationen zur Aktivierung und Konfiguration von Enhanced Scanning in Amazon ECR finden Sie unter Enhanced Scanning im Amazon ECR-Benutzerhandbuch.

Verwendung der Integration in einer Umgebung mit mehreren Konten

Wenn Sie Mitglied in einer Umgebung mit mehreren Konten sind, können Sie das erweiterte Scannen über Amazon ECR aktivieren. Nach der Aktivierung kann es jedoch nur von Ihrem delegierten Amazon Inspector-Administrator deaktiviert werden. Wenn es deaktiviert ist, kehrt es zum normalen Scannen zurück. Weitere Informationen finden Sie unter Amazon Inspector deaktivieren.

Amazon Inspector Inspector-Integration mit AWS Security Hub CSPM

Security Hub CSPM bietet einen umfassenden Überblick über Ihren Sicherheitsstatus in. AWS Auf diese Weise können Sie Ihre Umgebung anhand von Industriestandards und Best Practices im Bereich Sicherheit überprüfen. Security Hub CSPM sammelt Sicherheitsdaten von AWS Konten, Diensten und unterstützten Produkten. Sie können diese Informationen verwenden, um Sicherheitstrends zu analysieren und Sicherheitsprobleme zu identifizieren. Wenn Sie die Amazon Inspector-Integration mit Security Hub CSPM aktivieren, kann Amazon Inspector Ergebnisse an Security Hub CSPM senden, und Security Hub CSPM kann diese Ergebnisse im Rahmen Ihrer Sicherheitslage analysieren.

Security Hub CSPM verfolgt Sicherheitsprobleme als Ergebnisse. Einige Ergebnisse können auf Sicherheitsprobleme zurückzuführen sein, die in anderen AWS Diensten oder Produkten von Drittanbietern festgestellt wurden. Security Hub CSPM verwendet eine Reihe von Regeln, um Sicherheitsprobleme zu erkennen und Ergebnisse zu generieren, und stellt Tools bereit, mit denen Sie Ergebnisse verwalten können. Security Hub CSPM archiviert die Ergebnisse von Amazon Inspector, sobald die Ergebnisse in Amazon Inspector geschlossen wurden. Sie können auch eine Historie Ihrer Ergebnisse und Details zu den Ergebnissen einsehen sowie den Status einer Untersuchung zu einem Ergebnis verfolgen.

Security Hub CSPM verarbeitet Ergebnisse im AWS Security Finding Format (ASFF). Dieses Format enthält Details wie eindeutige Identifikatoren, Schweregrade, betroffene Ressourcen, Hinweise zur Problembehebung, Workflow-Status und Kontextinformationen.

> **Note**
>
> Von Amazon Inspector Code Security generierte Sicherheitsergebnisse sind für diese Integration nicht verfügbar. Sie können jedoch in der Amazon Inspector-Konsole und über die Amazon Inspector-API auf diese speziellen Ergebnisse zugreifen.

Themen

- Ergebnisse von Amazon Inspector anzeigen in AWS Security Hub CSPM
- Aktivierung und Konfiguration der Amazon Inspector Inspector-Integration mit Security Hub CSPM
- Aktivierung von Amazon Inspector über Security Hub CSPM mithilfe der Unternehmensrichtlinie
- Deaktivierung des Flusses von Ergebnissen aus einer Integration
- Sicherheitskontrollen für Amazon Inspector in Security Hub CSPM anzeigen

Ergebnisse von Amazon Inspector anzeigen in AWS Security Hub CSPM

Sie können die Ergebnisse von Amazon Inspector Classic und Amazon Inspector im Security Hub CSPM einsehen.

> **Note**
>
> Um nur nach Ergebnissen von Amazon Inspector `"aws/inspector/ProductVersion": "2"` zu filtern, fügen Sie es der Filterleiste hinzu. Dieser Filter schließt Amazon Inspector Classic-Ergebnisse aus dem Security Hub CSPM-Dashboard aus.

Beispiel für ein Ergebnis von Amazon Inspector

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:inspector2:us-east-1:123456789012:finding/FINDING_ID",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/inspector",
  "ProductName": "Inspector",
  "CompanyName": "Amazon",
  "Region": "us-east-1",
  "GeneratorId": "AWSInspector",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Vulnerabilities/CVE"
  ],
  "FirstObservedAt": "2023-01-31T20:25:38Z",
  "LastObservedAt": "2023-05-04T18:18:43Z",
  "CreatedAt": "2023-01-31T20:25:38Z",
  "UpdatedAt": "2023-05-04T18:18:43Z",
  "Severity": {
    "Label": "HIGH",
```

```
    "Normalized": 70
   },
   "Title": "CVE-2022-34918 - kernel",
   "Description": "An issue was discovered in the Linux kernel through 5.18.9. A type
  confusion bug in nft_set_elem_init (leading to a buffer overflow) could be used by a
  local attacker to escalate privileges, a different vulnerability than CVE-2022-32250.
  (The attacker can obtain root access, but must start with an unprivileged user
  namespace to obtain CAP_NET_ADMIN access.) This can be fixed in nft_setelem_parse_data
  in net/netfilter/nf_tables_api.c.",
   "Remediation": {
     "Recommendation": {
       "Text": "Remediation is available. Please refer to the Fixed version in the
  vulnerability details section above. For detailed remediation guidance for each of the
  affected packages, refer to the vulnerabilities section of the detailed finding JSON."
     }
   },
   "ProductFields": {
     "aws/inspector/FindingStatus": "ACTIVE",
     "aws/inspector/inspectorScore": "7.8",
     "aws/inspector/resources/1/resourceDetails/awsEc2InstanceDetails/platform":
  "AMAZON_LINUX_2",
     "aws/inspector/ProductVersion": "2",
     "aws/inspector/instanceId": "i-0f1ed287081bdf0fb",
     "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/inspector/
  arn:aws:inspector2:us-east-1:123456789012:finding/FINDING_ID",
     "aws/securityhub/ProductName": "Inspector",
     "aws/securityhub/CompanyName": "Amazon"
   },
   "Resources": [
     {
       "Type": "AwsEc2Instance",
       "Id": "arn:aws:ec2:us-east-1:123456789012:i-0f1ed287081bdf0fb",
       "Partition": "aws",
       "Region": "us-east-1",
       "Tags": {
         "Patch Group": "SSM",
         "Name": "High-SEv-Test"
       },
       "Details": {
         "AwsEc2Instance": {
           "Type": "t2.micro",
           "ImageId": "ami-0cff7528ff583bf9a",
           "IpV4Addresses": [
             "52.87.229.97",
```

```
            "172.31.57.162"
          ],
          "KeyName": "ACloudGuru",
          "IamInstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
          "VpcId": "vpc-a0c2d7c7",
          "SubnetId": "subnet-9c934cb1",
          "LaunchedAt": "2022-07-26T21:49:46Z"
        }
      }
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "Vulnerabilities": [
    {
      "Id": "CVE-2022-34918",
      "VulnerablePackages": [
        {
          "Name": "kernel",
          "Version": "5.10.118",
          "Epoch": "0",
          "Release": "111.515.amzn2",
          "Architecture": "X86_64",
          "PackageManager": "OS",
          "FixedInVersion": "0:5.10.130-118.517.amzn2",
          "Remediation": "yum update kernel"
        }
      ],
      "Cvss": [
        {
          "Version": "2.0",
          "BaseScore": 7.2,
          "BaseVector": "AV:L/AC:L/Au:N/C:C/I:C/A:C",
          "Source": "NVD"
        },
        {
          "Version": "3.1",
          "BaseScore": 7.8,
          "BaseVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H",
          "Source": "NVD"
```

```
        },
        {
          "Version": "3.1",
          "BaseScore": 7.8,
          "BaseVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H",
          "Source": "NVD",
          "Adjustments": []
        }
      ],
      "Vendor": {
        "Name": "NVD",
        "Url": "https://nvd.nist.gov/vuln/detail/CVE-2022-34918",
        "VendorSeverity": "HIGH",
        "VendorCreatedAt": "2022-07-04T21:15:00Z",
        "VendorUpdatedAt": "2022-10-26T17:05:00Z"
      },
      "ReferenceUrls": [
        "https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net.git/commit/?
id=7e6bc1f6cabcd30aba0b11219d8e01b952eacbb6",
        "https://lore.kernel.org/netfilter-devel/cd9428b6-7ffb-dd22-d949-
d86f4869f452@randorisec.fr/T/",
        "https://www.debian.org/security/2022/dsa-5191"
      ],
      "FixAvailable": "YES"
    }
  ],
  "FindingProviderFields": {
    "Severity": {
      "Label": "HIGH"
    },
    "Types": [
      "Software and Configuration Checks/Vulnerabilities/CVE"
    ]
  },
  "ProcessedAt": "2023-05-05T20:28:38.822Z"
}
```


Aktivierung und Konfiguration der Amazon Inspector Inspector-Integration mit Security Hub CSPM

Sie können die Amazon Inspector Inspector-Integration mit aktivieren, AWS Security Hub CSPM indem Sie Security Hub CSPM aktivieren. Nachdem Sie Security Hub CSPM aktiviert haben, AWS

Security Hub CSPM wird die Amazon Inspector-Integration mit automatisch aktiviert und Amazon Inspector beginnt, alle Ergebnisse unter Verwendung des Security Finding Formats (ASFF) an AWS Security Hub CSPM zu senden.

Aktivierung von Amazon Inspector über Security Hub CSPM mithilfe der Unternehmensrichtlinie

Sie können die Amazon Inspector Inspector-Aktivierung in Ihrer gesamten AWS Organizations mithilfe von Unternehmensrichtlinien direkt von der Security Hub CSPM-Konsole aus verwalten. Dieser zentralisierte Ansatz ermöglicht es Ihnen, Amazon Inspector über die Richtlinienverwaltung auf Organisationsebene das gleichzeitige Scannen mehrerer Konten zu ermöglichen.

Detaillierte Anweisungen zur Verwaltung der Amazon Inspector Inspector-Aktivierung über Security Hub CSPM mithilfe von Unternehmensrichtlinien finden Sie unter Verwaltung delegierter Administratorkonten für Security Hub CSPM im Benutzerhandbuch.AWS Security Hub CSPM

Deaktivierung des Flusses von Ergebnissen aus einer Integration

Um zu verhindern, dass Amazon Inspector Ergebnisse an Security Hub CSPM sendet, können Sie die Security Hub CSPM-Konsole oder API verwenden und... AWS CLI

Sicherheitskontrollen für Amazon Inspector in Security Hub CSPM anzeigen

Security Hub CSPM analysiert Ergebnisse von unterstützten Produkten AWS und Produkten von Drittanbietern und führt automatisierte und kontinuierliche Sicherheitsprüfungen anhand von Regeln durch, um eigene Ergebnisse zu generieren. Die Regeln werden durch Sicherheitskontrollen dargestellt, anhand derer Sie feststellen können, ob die Anforderungen eines Standards erfüllt werden.

Amazon Inspector verwendet Sicherheitskontrollen, um zu überprüfen, ob die Funktionen von Amazon Inspector aktiviert sind oder aktiviert werden sollten. Nachstehend sind einige dieser Features aufgeführt:

- Amazon EC2-Scannen
- Amazon ECR-Scannen
- Lambda-Standardabtastung
- Scannen von Lambda-Code

Weitere Informationen finden Sie unter Amazon Inspector-Steuerelemente im AWS Security Hub CSPM Benutzerhandbuch.

Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector

Amazon Inspector kann Softwareanwendungen scannen, die auf folgenden Geräten installiert sind:

- Instances von Amazon Elastic Compute Cloud (Amazon EC2)

  > **Note**
  >
  > Für Amazon EC2 EC2-Instances kann Amazon Inspector nach Paketschwachstellen in Betriebssystemen suchen, die agentenbasiertes Scannen unterstützen. Amazon Inspector kann auch nach Paketschwachstellen in Betriebssystemen und Programmiersprachen suchen, die Hybrid-Scans unterstützen. Amazon Inspector sucht nicht nach Schwachstellen in der Toolchain. Die Version des Programmiersprachen-Compilers, der zur Erstellung der Anwendung verwendet wurde, führt zu diesen Sicherheitslücken.

- Container-Images, die in Amazon Elastic Container Registry (Amazon ECR) -Repositorys gespeichert sind

  > **Note**
  >
  > Bei ECR-Container-Images kann Amazon Inspector nach Sicherheitslücken in Betriebssystemen und Programmiersprachenpaketen suchen. Amazon Inspector unterstützt auch gehärtete Images, die von ChainguardMinimus, EchoDocker, und bereitgestellt Red Hat werden. Amazon Inspector sucht nicht nach Schwachstellen in der Toolchain in Rust — die Version des Programmiersprachen-Compilers, der zur Erstellung der Anwendung verwendet wurde, führt zu diesen Sicherheitslücken.

- AWS Lambda Funktionen

  > **Note**
  >
  > Bei Lambda-Funktionen kann Amazon Inspector nach Sicherheitslücken in Programmiersprachenpaketen und Code-Schwachstellen suchen. Amazon Inspector sucht nicht nach Schwachstellen in der Toolchain. Die Version des Programmiersprachen-

Compilers, der zur Erstellung der Anwendung verwendet wurde, führt zu diesen Sicherheitslücken.

Wenn Amazon Inspector Ressourcen scannt, bezieht Amazon Inspector mehr als 50 Datenfeeds, um Ergebnisse für häufig auftretende Sicherheitslücken und Risiken (CVEs) zu generieren. Zu diesen Quellen gehören beispielsweise Sicherheitsempfehlungen von Anbietern, Datenfeeds und Feeds mit Bedrohungsinformationen sowie die National Vulnerability Database (NVD) und MITRE. Amazon Inspector aktualisiert Sicherheitslückendaten aus Quell-Feeds mindestens einmal täglich.

Damit Amazon Inspector eine Ressource scannen kann, muss auf der Ressource ein unterstütztes Betriebssystem oder eine unterstützte Programmiersprache ausgeführt werden. In den Themen in diesem Abschnitt sind die Betriebssysteme, Programmiersprachen und Laufzeiten aufgeführt, die Amazon Inspector für verschiedene Ressourcen und Scantypen unterstützt. Sie listen auch eingestellte Betriebssysteme auf.

Note

Amazon Inspector kann nur eingeschränkten Support für ein Betriebssystem anbieten, nachdem ein Anbieter den Support für das Betriebssystem eingestellt hat.

Themen

- Unterstützte Betriebssysteme
- Eingestellte Betriebssysteme
- Unterstützte Programmiersprachen
- Unterstützte Laufzeiten

Unterstützte Betriebssysteme

In diesem Abschnitt sind die Betriebssysteme aufgeführt, die Amazon Inspector unterstützt. In jeder Tabelle sind die Beratungsquelle und der Beratungsstatus für jedes Betriebssystem angegeben.

In der Spalte „Beratungsstatus“ werden die Status der Lieferantenempfehlungen aufgeführt, anhand derer Amazon Inspector Ergebnisse für ein Betriebssystem generiert. Die Spalte kann die folgenden Werte enthalten:

Fixed

Der Hersteller hat einen Fix für die Sicherheitsanfälligkeit veröffentlicht.

Öffnen

Der Anbieter hat die Sicherheitsanfälligkeit bestätigt, aber keinen Fix veröffentlicht.

Aktiv

Der Anbieter verfolgt die Sicherheitsanfälligkeit, und eine Lösung ist noch nicht verfügbar.

Ausstehend

Der Anbieter bereitet einen Fix vor, der noch nicht veröffentlicht wurde.

Für einige Betriebssysteme enthält die Advisory-Quelle zusätzliche Repositorys:

RHEL

Die Beratungsquelle umfasst das BaseOS-Repository und die EUS-, E2S- und E4S-Repositorien. Weitere Informationen finden Sie auf der Website unter Red Hat Enterprise Linux Life Cycle. Red Hat

Ubuntu

Die Beratungsquelle umfasst Ubuntu Pro (esm-infraundesm-apps). Weitere Informationen finden Sie unter Ubuntu-Sicherheitshinweise und erweiterte Sicherheitswartung auf der Ubuntu Website.

Unterstützte Betriebssysteme: Amazon EC2-Scanning

In der folgenden Tabelle sind die Betriebssysteme aufgeführt, die Amazon Inspector für das Scannen von Amazon EC2 EC2-Instances unterstützt. Sie gibt die Beratungsquelle und den Beratungsstatus für jedes Betriebssystem an und gibt an, welche Betriebssysteme agentenbasiertes Scannen und agentenloses Scannen unterstützen.

Wenn Sie die agentenbasierte Scanmethode verwenden, konfigurieren Sie den SSM-Agent so, dass er kontinuierliche Scans auf allen geeigneten Instanzen durchführt. Amazon Inspector empfiehlt, eine Version des SSM-Agenten zu konfigurieren, die höher als 3.2.2086.0 ist. Weitere Informationen finden Sie unter Arbeiten mit dem SSM-Agenten im Amazon EC2 Systems Manager Manager-Benutzerhandbuch.

Erkennungen von Linux-Betriebssystemen werden nur für das Standard-Paketmanager-Repository (rpm und dpkg) unterstützt. Zu den Erkennungen gehören keine Anwendungen von Drittanbietern, Repositorys mit erweitertem Support oder optionale Repositorys (Anwendungsstreams), sofern unten nicht anders angegeben. Amazon Inspector scannt den laufenden Kernel auf Sicherheitslücken. Bei einigen Betriebssystemen ist beispielsweise ein Neustart erforderlichUbuntu, damit Upgrades in aktiven Ergebnissen angezeigt werden.

Definitionen der Werte für den Status der Empfehlungen und Einzelheiten zu den Quellen der Empfehlung finden Sie unterUnterstützte Betriebssysteme.

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Unterstützung für agentenloses Scannen	Agent-based Scan-Unterstützung
AlmaLinux	8	CVE-Errata	Fixed	Ja	Ja
AlmaLinux	9	CVE-Fehler	Fixed	Ja	Ja
AlmaLinux	10	CVE-Fehler	Fixed	Ja	Ja
Amazon Linux 2023 (AL2023)	AL2023	CVE-Fehler	Fixed	Ja	Ja
Flaschenrakete	1.7.0 und später	CVE-Errata	Fixed	Ja	Ja
Debian-Server (Bücherwurm)	12	DSA-HÖHLE	Behoben, offen	Ja	Ja
Debian-Server (Trixie)	13	DSA-HÖHLE	Behoben, offen	Ja	Ja
Fedora	43	CVE Errata	Fixed	Ja	Ja
Fedora	44	CVE Errata	Fixed	Ja	Ja

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Unterstützung für agentenloses Scannen	Agent-based Scan-Unterstützung
Oracle Linux	8	CVE-Errata	Fixed	Ja	Ja
Oracle Linux	9	CVE-Fehler	Fixed	Ja	Ja
Oracle Linux	10	CVE-Fehler	Fixed	Ja	Ja
Red Hat Enterprise Linux (RHEL)	8	RHEL VEX CVE	Fixed	Ja	Ja
Red Hat Enterprise Linux (RHEL)	9	RHEL VEX CVE	Fixed	Ja	Ja
Red Hat Enterprise Linux (RHEL)	10	RHEL VEX CVE	Fixed	Ja	Ja
Rocky Linux	8	CVE-Fehler	Fixed	Ja	Ja
Rocky Linux	9	CVE-Fehler	Fixed	Ja	Ja
Rocky Linux	10	CVE-Fehler	Fixed	Ja	Ja
SUSE Linux Enterprise Server (SLES)	15.7	CVE-Fehler	Fixed	Ja	Ja
SUSE Linux Enterprise Server (SLES)	16.0	CVE-Fehler	Fixed	Ja	Ja

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Unterstützung für agentenloses Scannen	Agent-based Scan-Unterstützung
Ubuntu (Bionisch)	18,04	SONNE	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Focal)	20.04	SONNE	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Jammy)	22,04	SONNE	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Noble)	24.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Resolute)	26.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Windows Server	2016	MSKB	Fixed	Ja	Ja
Windows Server	2019	MSKB	Fixed	Ja	Ja
Windows Server	2022	MSKB	Fixed	Ja	Ja
Windows Server	2025	MSKB	Fixed	Ja	Ja
macOS (Mojave)	10.14	APPLE-SA	Fixed	Nein	Ja

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Unterstützung für agentenloses Scannen	Agent-based Scan-Unterstützung
macOS (Catalina)	10.15	APPLE-SA	Fixed	Nein	Ja
macOS (Big Sur)	11	APPLE-SA	Fixed	Nein	Ja
macOS (Monterey)	12	APPLE-SA	Fixed	Nein	Ja
macOS (Ventura)	13	APPLE-SA	Fixed	Nein	Ja
macOS (Sonoma)	14	APPLE-SA	Fixed	Nein	Ja
macOS (Sequoia)	15	APPLE-SA	Fixed	Nein	Ja
macOS (Tahoe)	26	APPLE-SA	Fixed	Nein	Ja

Unterstützte Betriebssysteme: Amazon ECR-Scannen mit Amazon Inspector

In der folgenden Tabelle sind die Betriebssysteme aufgeführt, die Amazon Inspector für das Scannen von Container-Images in Amazon ECR-Repositorys unterstützt. Außerdem werden die Beratungsquelle und der Beratungsstatus für jedes Betriebssystem angegeben.

Definitionen der Werte für den Status der Empfehlung und Einzelheiten zu den Quellen für die Empfehlung finden Sie unterUnterstützte Betriebssysteme.

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Einfaches Scannen	Erweitertes Scannen
AlmaLinux	8	CVE Errata	Fixed	Ja	Ja
AlmaLinux	9	CVE Errata	Fixed	Ja	Ja
AlmaLinux	10	CVE Errata	Fixed	Ja	Ja
Alpine Linux	3.21	CVE Errata	Fixed	Ja	Ja
Alpine Linux	3.22	CVE Errata	Fixed	Ja	Ja
Alpine Linux	3.23	CVE Errata	Fixed	Ja	Ja
Alpine Linux	3.24	CVE Errata	Fixed	Ja	Ja
Amazon Linux 2023 (AL2023)	AL2023	CVE Errata	Fixed	Ja	Ja
Azure Linux	3	CVE Errata	Fixed	Ja	Ja
BusyBox	–	MITRE CVE	Fixed	Ja	Ja
Chainguard	–	OSV	Fixed	Ja	Ja
Debian Server (Bookworm)	12	DSA CVE	Behoben, Offen	Ja	Ja
Debian Server (Trixie)	13	DSA CVE	Behoben, Offen	Ja	Ja
Echo	2	CVE Errata	Fixed	Ja	Ja
Fedora	43	CVE Errata	Fixed	Ja	Ja
Fedora	44	CVE Errata	Fixed	Ja	Ja

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Einfaches Scannen	Erweitertes Scannen
Hummingbird OS	–	RHEL VEX CVE	Fixed	Ja	Nein
MinimOS	–	CVE Errata	Fixed	Ja	Ja
Oracle Linux	8	CVE Errata	Fixed	Ja	Ja
Oracle Linux	9	CVE Errata	Fixed	Ja	Ja
Oracle Linux	10	CVE Errata	Fixed	Ja	Ja
Photon OS	4	CVE Errata	Fixed	Ja	Ja
Photon OS	5	CVE Errata	Fixed	Ja	Ja
Red Hat Enterprise Linux (RHEL)	8	RHEL VEX CVE	Fixed	Ja	Ja
Red Hat Enterprise Linux (RHEL)	9	RHEL VEX CVE	Fixed	Ja	Ja
Red Hat Enterprise Linux (RHEL)	10	RHEL VEX CVE	Fixed	Ja	Ja
Rocky Linux	8	CVE Errata	Fixed	Ja	Ja
Rocky Linux	9	CVE Errata	Fixed	Ja	Ja
Rocky Linux	10	CVE Errata	Fixed	Ja	Ja
SUSE Linux Enterprise Server (SLES)	15.7	CVE Errata	Fixed	Ja	Ja

Betriebssystem	Version	Quelle der Beratung	Status der Beratung	Einfaches Scannen	Erweitertes Scannen
SUSE Linux Enterprise Server (SLES)	16.0	CVE Errata	Fixed	Ja	Ja
Ubuntu (Bionic)	18.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Focal)	20.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Jammy)	22.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Noble)	24.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Ubuntu (Resolute)	26.04	USN	Behoben, Aktiv, Ausstehend	Ja	Ja
Windows Server	2019	MSKB	Fixed	Ja	Nein
Windows Server	2022	MSKB	Fixed	Ja	Nein
Windows Server	2025	MSKB	Fixed	Ja	Nein
Wolfi	–	OSV	Fixed	Ja	Ja

Unterstützte Betriebssysteme: CIS-Scanning

In der folgenden Tabelle sind die Betriebssysteme aufgeführt, die Amazon Inspector für CIS-Scans unterstützt. Es spezifiziert auch die CIS-Benchmark-Version für jedes Betriebssystem.

> **Note**
>
> CIS-Standards sind für x86_64-Betriebssysteme vorgesehen. Einige Prüfungen werden möglicherweise nicht ausgewertet oder geben ungültige Anweisungen zur Behebung von Ressourcen zurück. ARM-based

Betriebssystem	Version	CIS-Benchmark-Version
Amazon Linux 2023	AL2023	1.0.0
Red Hat Enterprise Linux (RHEL)	8	4.0.0
Red Hat Enterprise Linux (RHEL)	9	2.0.0
Rocky Linux	8	3.0.0
Rocky Linux	9	2.0.0
SUSE-Linux-Enterprise-Server	15	2.0.1
Ubuntu (fokal)	20.04	3.0.0
Ubuntu (Jammy)	22,04	3.0.0
Ubuntu (edel)	24,04	2.0.0
Windows Server	2016	4.0.0
Windows Server	2019	5.0.0
Windows Server	2022	5.0.0

Betriebssystem	Version	CIS-Benchmark-Version
Windows Server	2025	2.0.0

Unterstützte Betriebssysteme: Amazon Inspector Scan API

In der folgenden Tabelle sind die unterstützten Betriebssysteme für die Amazon Inspector Scan API aufgeführt. Weitere Informationen finden Sie ScanSbomin der Amazon Inspector V2 API-Referenz.

Betriebssystem	Version
AlmaLinux	8
AlmaLinux	9
AlmaLinux	10
Alpine Linux	3.21
Alpine Linux	3,22
Alpine Linux	3,23
Alpine Linux	3,24
Amazon Linux 2023 (AL2023)	AL2023
Azure Linux	3
Bottlerocket	–
BusyBox	1,36,0+
Chainguard	–
CleanStart	–
Debian Server (Bookworm)	12
Debian Server (Trixie)	13

Betriebssystem	Version
Debian Sid	–
Echo	2
Fedora	43
Fedora	44
Hummingbird OS	–
macOS	11
MinimOS	–
Oracle Linux	8
Oracle Linux	9
Oracle Linux	10
Photon OS	4
Photon OS	5
Red Hat Enterprise Linux (RHEL)	8
Red Hat Enterprise Linux (RHEL)	9
Red Hat Enterprise Linux (RHEL)	10
Rocky Linux	8
Rocky Linux	9
Rocky Linux	10
SUSE Linux Enterprise Server (SLES)	15,7
SUSE Linux Enterprise Server (SLES)	16,0

Betriebssystem	Version
Ubuntu (Bionic)	18,04
Ubuntu (Focal)	20,04
Ubuntu (Jammy)	22,04
Ubuntu (Noble)	24,04
Ubuntu (Resolute)	26,04
Ubuntu (Stonking)	26,10
Wolfi	–
Windows Server	2016
Windows Server	2019
Windows Server	2022
Windows Server	2025

Eingestellte Betriebssysteme

In der folgenden Tabelle sind Betriebssysteme aufgeführt, die eingestellt wurden, und wann sie eingestellt wurden.

Auch wenn Amazon Inspector keine vollständige Unterstützung für eingestellte Betriebssysteme bietet, scannt Amazon Inspector weiterhin Amazon EC2 EC2-Instances und Amazon ECR-Container-Images, auf denen sie ausgeführt werden. Aus Sicherheitsgründen generiert Amazon Inspector einen KRITISCHEN Befund für Ressourcen, die ein eingestelltes Betriebssystem verwenden, und empfiehlt, auf eine unterstützte Version umzusteigen. Ergebnisse, die Amazon Inspector für eingestellte Betriebssysteme generiert, sollten nur zu Informationszwecken verwendet werden.

Gemäß den Herstellerrichtlinien erhalten ausgelaufene Betriebssysteme keine Patch-Updates oder Sicherheitsempfehlungen mehr. Anbieter können auch bestehende Sicherheitsempfehlungen und Sicherheitswarnungen aus ihren Feeds für Betriebssysteme entfernen, für die der Standardsupport

ausläuft. Aus diesem Grund generiert Amazon Inspector 12 Monate nach den unten aufgeführten Terminen keine Ergebnisse mehr für eingestellte Betriebssysteme.

Betriebssystem	Version	Nicht mehr angeboten
Alpine Linux	3.2	1. Mai 2017
Alpine Linux	3.3	1. November 2017
Alpine Linux	3.4	1. Mai 2018
Alpine Linux	3.5	1. November 2018
Alpine Linux	3.6	1. Mai 2019
Alpine Linux	3.7	1. November 2019
Alpine Linux	3.8	1. Mai 2020
Alpine Linux	3.9	1. November 2020
Alpine Linux	3,10	1. Mai 2021
Alpine Linux	3,11	1. November 2021
Alpine Linux	3,12	01.Mai 2022
Alpine Linux	3.13	1. November 2022
Alpine Linux	3,14	1. Mai 2023
Alpine Linux	3,15	1. November 2023
Alpine Linux	3,16	23. Mai 2024
Alpine Linux	3,17	22. November 2024
Alpine Linux	3,18	9. Mai 2025
Alpine Linux	3.19	1. November 2025
Alpine Linux	3,20	1. April 2026

Betriebssystem	Version	Nicht mehr angeboten
Amazon Linux (AL1)	2012	31. Dezember 2021
Amazon Linux 2 (AL2)	AL 2	30. Juni 2026
CentOS Linux (CentOS)	7	30. Juni 2024
CentOS Linux (CentOS)	8	31. Dezember 2021
Debian-Server (Jessie)	8	30. Juni 2020
Debian-Server (Stretch)	9	30. Juni 2022
Debian-Server (Buster)	10	30. Juni 2024
Debian-Server (Bullseye)	11	31. August 2026
Fedora	33	30. November 2021
Fedora	34	7. Juni 2022
Fedora	35	13. Dezember 2022
Fedora	36	16. Mai 2023
Fedora	37	15. Dezember 2023
Fedora	38	21. Mai 2024
Fedora	39	26. November 2024
Fedora	40	13. Mai 2025
Fedora	41	19. November 2025
Fedora	42	13. Mai 2026
openSUSE Leap	15.2	1. Dezember 2021
openSUSE Leap	15.3	01. Dezember 2022

Betriebssystem	Version	Nicht mehr angeboten
openSUSE Leap	15.4	07. Dezember 2023
openSUSE Leap	15,5	31. Dezember 2024
openSUSE Leap	15.6	30. April 2026
Oracle Linux	6	1. März 2021
Oracle Linux	7	31. Dezember 2024
Photon OS	2	2. Dezember 2021
Photon Betriebssystem	3	1. März 2024
Red Hat Enterprise Linux (RHEL)	6	30. Juni 2020
Red Hat Enterprise Linux (RHEL)	7	30. Juni 2024
SUSE Linux Enterprise Server (SLES)	12	30. Juni 2016
SUSE Linux Enterprise Server (SLES)	12.1	31. Mai 2017
SUSE Linux Enterprise Server (SLES)	12.2	31. März 2018
SUSE Linux Enterprise Server (SLES)	12.3	30. Juni 2019
SUSE Linux Enterprise Server (SLES)	12.4	30. Juni 2020
SUSE Linux Enterprise Server (SLES)	12,5	31. Oktober 2024
SUSE Linux Enterprise Server (SLES)	15	31. Dezember 2019

Betriebssystem	Version	Nicht mehr angeboten
SUSE Linux Enterprise Server (SLES)	15,1	31. Januar 2021
SUSE Linux Enterprise Server (SLES)	15.2	31. Dezember 2021
SUSE Linux Enterprise Server (SLES)	15.3	31. Dezember 2022
SUSE Linux Enterprise Server (SLES)	15,4	31. Dezember 2023
SUSE Linux Enterprise Server (SLES)	15,5	31. Dezember 2024
SUSE Linux Enterprise Server (SLES)	15,6	31. Dezember 2025
Ubuntu (Präzise)	12,04	28. April 2017
Ubuntu (Vertrauenswürdig)	14.04	1. April 2024
Ubuntu (Xenial)	16,04	1. April 2026
Ubuntu (Groovy)	20,10	22. Juli 2021
Ubuntu (Hirsute)	21,04	20. Januar 2022
Ubuntu (Impish)	21.10	31. Juli 2022
Ubuntu (Kinetisch)	22,10	20. Juli 2023
Ubuntu (Mond)	23,04	25. Januar 2024
Ubuntu (Mantik)	23,10	11. Juli 2024
Ubuntu (Oracular)	24,10	10. Juli 2025
Ubuntu (Mutig)	25,04	15. Januar 2026

Betriebssystem	Version	Nicht mehr angeboten
Ubuntu (Questen)	25.10	9. Juli 2026
Windows Server	2012	10. Oktober 2023
Windows Server	2012 R2	10. Oktober 2023

Unterstützte Programmiersprachen

In diesem Abschnitt sind die Programmiersprachen aufgeführt, die Amazon Inspector unterstützt.

Unterstützte Programmiersprachen: Agentenloses Scannen mit Amazon EC2

Amazon Inspector unterstützt derzeit die folgenden Programmiersprachen bei der Durchführung von agentenlosen Scans auf geeigneten Amazon EC2 EC2-Instances. Weitere Informationen finden Sie unter Agentloses Scannen.

> **Note**
>
> Amazon Inspector sucht in Go und Rust nicht nach Schwachstellen in der Toolchain. Die Version des Programmiersprachen-Compilers, der zur Erstellung der Anwendung verwendet wurde, führt zu diesen Sicherheitslücken.

- C#
- Go
- Java
- JavaScript
- PHP
- Python
- Ruby
- Rust

Unterstützte Programmiersprachen: Amazon Inspector VM Scanner

Der Amazon Inspector VM Scanner unterstützt dieselben Programmiersprachen wie das agentenlose Scannen von Amazon EC2. Eine Liste der unterstützten Sprachen finden Sie unter. Unterstützte Programmiersprachen: Agentenloses Scannen mit Amazon EC2

Unterstützte Programmiersprachen: Amazon Inspector SSM-Plugin

Amazon Inspector unterstützt derzeit die folgenden Programmiersprachen bei der Durchführung von Deep Inspection-Scans auf Amazon EC2 EC2-Linux-Instances mithilfe des Amazon Inspector SSM-Plug-ins. Deep Inspection über das Amazon Inspector SSM-Plugin unterstützt eine Teilmenge der Programmiersprachen, die vom Amazon Inspector VM Scanner unterstützt werden. Weitere Informationen finden Sie unter Eingehende Inspektion von Amazon Inspector für Linux-based Amazon EC2 EC2-Instances.

Die hier aufgeführten Sprachen gelten für die Tiefeninspektion über das Amazon Inspector SSM-Plugin für Linux Instances. Informationen zur Tiefeninspektion mit Enhanced EC2 Scanning (dem Amazon Inspector VM Scanner) finden Sie unter Amazon Inspector VM Scanner. Der Amazon Inspector VM Scanner unterstützt LinuxWindows, und macOS Instances.

- Java(Archivformate .ear, .jar, .par und .war)
- JavaScript
- Python

Amazon Inspector verwendet Systems Manager Distributor, um das Plug-in für eine gründliche Inspektion Ihrer Amazon EC2 EC2-Instance bereitzustellen.

> **Note**
>
> Die Tiefeninspektion wird für Bottlerocket-Betriebssysteme nicht unterstützt.

Um Deep Inspection-Scans durchführen zu können, müssen Systems Manager Distributor und Amazon Inspector Ihr Amazon EC2 EC2-Instance-Betriebssystem unterstützen. Informationen zu den unterstützten Betriebssystemen in Systems Manager Distributor finden Sie unter Unterstützte Paketplattformen und Architekturen im Systems Manager Manager-Benutzerhandbuch.

Unterstützte Programmiersprachen: Amazon ECR Scanning

Amazon Inspector unterstützt derzeit die folgenden Programmiersprachen beim Scannen von Container-Images in Amazon ECR-Repositorys:

> **Note**
>
> Amazon Inspector sucht nicht nach Schwachstellen in Rust der Toolchain. Die Version des Programmiersprachen-Compilers, der zur Erstellung der Anwendung verwendet wurde, führt zu diesen Sicherheitslücken. Für Python Java Anwendungen, die Chainguard Bibliotheken verwenden, erkennt Amazon Inspector die rückportierten Sicherheitskorrekturen und schließt sie aus den Ergebnissen aus. Amazon Inspector erkennt auch die rückportierten Sicherheitskorrekturen für PythonJava, und JavaScript Anwendungen, die Echo Bibliotheken verwenden. Weitere Informationen finden Sie unter ChainguardBibliotheken auf der Chainguard Website und EchoBibliotheken auf der Echo Website.

- C#
- Go
- GoToolchain
- Java(einschließlich Chainguard Echo Bibliotheken)
- JavaJDK
- JavaScript(einschließlich Echo Bibliotheken)
- PHP
- Python(einschließlich Chainguard Echo Bibliotheken)
- Ruby
- Rust

Unterstützte Laufzeiten

In diesem Abschnitt sind die Laufzeiten aufgeführt, die Amazon Inspector unterstützt.

Unterstützte Laufzeiten: Amazon Inspector Lambda Standard-Scanning

Das Standard-Scannen von Amazon Inspector Lambda unterstützt derzeit die folgenden Laufzeiten für die Programmiersprachen, die beim Scannen von Lambda-Funktionen auf Sicherheitslücken in Softwarepaketen von Drittanbietern verwendet werden können:

> **Note**
>
> Amazon Inspector sucht nicht nach Schwachstellen in Rust der Toolchain. Die Version des Programmiersprachen-Compilers, der zur Erstellung der Anwendung verwendet wurde, führt zu diesen Sicherheitslücken.

- Go
  - go1.x
- Java
  - java8
  - java8.al2
  - java11
  - java17
  - java21
  - java25
- .NET
  - .NET 6
  - .NET 8
  - .NET 10
- Node.js
  - nodejs12.x
  - nodejs14.x
  - nodejs16.x
  - nodejs18.x
  - nodejs20.x
  - nodejs22.x

  - nodejs24.x
- Python
  - python3.7
  - python3.8
  - python3.9
  - python3.10
  - python3.11
  - python3.12
  - python3.13
  - python3.14
- Ruby
  - ruby2.7
  - ruby3.2
  - ruby3.3
- Custom runtimes
  - AL2
  - AL2023

Unterstützte Laufzeiten: Amazon Inspector Lambda-Code-Scanning

Das Lambda-Codescanning von Amazon Inspector unterstützt derzeit die folgenden Laufzeiten für die Programmiersprachen, die beim Scannen von Lambda-Funktionen auf Sicherheitslücken im Code verwendet werden können:

- Java
  - java8
  - java8.al2
  - java11
  - java17
  - java21
  - java25
- .NET

- .NET 6
- .NET 8

- Node.js
  - nodejs12.x
  - nodejs14.x
  - nodejs16.x
  - nodejs18.x
  - nodejs20.x
  - nodejs22.x
- Python
  - python3.7
  - python3.8
  - python3.9
  - python3.10
  - python3.11
  - python3.12
  - python3.13
  - python3.14
- Ruby
  - ruby2.7
  - ruby3.2
  - ruby3.3

Amazon Inspector deaktivieren

Sie können Amazon Inspector in der Amazon Inspector Inspector-Konsole oder mit der Amazon Inspector Inspector-API deaktivieren. Wenn Sie alle Scanarten für ein Konto deaktivieren, wird Amazon Inspector für dieses Konto automatisch deaktiviert.

Wenn Sie Amazon Inspector für ein Konto deaktivieren, werden alle Scanarten für dieses Konto deaktiviert. Darüber hinaus werden alle Amazon Inspector-Scaneinstellungen, einschließlich Filter, Unterdrückungsregeln und Ergebnisse, für das Konto gelöscht.

Wenn Sie Amazon Inspector EC2 Amazon-Scanning deaktivieren, löscht Amazon Inspector die folgenden SSM-Verknüpfungen:

- `InspectorDistributor-do-not-delete`
- `InspectorInventoryCollection-do-not-delete`
- `InvokeInspectorSsmPlugin-do-not-delete`. Darüber hinaus wird das Amazon Inspector SSM-Plugin, das über diese Verknüpfung installiert wurde, von all Ihren Windows Hosts entfernt. Weitere Informationen finden Sie unter EC2-Instance wird Windows gescannt.

> **Note**
>
> Sobald Sie Amazon Inspector deaktivieren, fallen keine Servicegebühren mehr an. Sie können Amazon Inspector jedoch jederzeit reaktivieren.

Informationen zum Deaktivieren von Scantypen für verschiedene Ressourcen finden Sie unter Deaktivieren eines Scantyps.

Voraussetzungen

Beachten Sie je nach Kontotyp Folgendes:

- Wenn es sich bei Ihrem Konto um ein eigenständiges Amazon Inspector-Konto handelt, können Sie Amazon Inspector jederzeit deaktivieren.
- Wenn es sich bei Ihrem Konto um ein Mitgliedskonto in einer Umgebung mit mehreren Konten handelt, können Sie Amazon Inspector nicht deaktivieren. Sie müssen sich an den delegierten Administrator Ihrer Organisation wenden, um Amazon Inspector zu deaktivieren.

- Wenn Sie der delegierte Administrator für eine Organisation sind, müssen Sie alle Ihre Mitgliedskonten trennen, bevor Sie Amazon Inspector deaktivieren.
- Wenn die Amazon Inspector Inspector-Aktivierung Ihres Kontos durch AWS Organizations Richtlinien verwaltet wird, können Sie richtlinienverwaltete Scantypen nicht über die Amazon Inspector Inspector-Konsole oder API deaktivieren. Um Amazon Inspector-Scantypen zu deaktivieren, müssen Sie die Unternehmensrichtlinie ändern, um sie explizit über die AWS Organizations Konsole oder API zu deaktivieren. Sie können Scantypen, die nicht durch Unternehmensrichtlinien verwaltet werden, über die Amazon Inspector Inspector-Konsole oder API deaktivieren.

> **Note**
>
> Wenn Sie Amazon Inspector als delegierter Administrator deaktivieren, deaktivieren Sie die automatische Aktivierungsfunktion für Ihre Organisation.

Deaktivierung von Amazon Inspector, der von Unternehmensrichtlinien verwaltet wird

Wenn Amazon Inspector in Ihren Konten durch AWS Organizations Richtlinien aktiviert ist, müssen Sie Inspector mithilfe der AWS Organizations Konsole oder API deaktivieren. Mitgliedskonten und delegierte Administratoren können richtlinienverwaltete Scantypen nicht über die Amazon Inspector Inspector-Konsole oder API deaktivieren.

So deaktivieren Sie Amazon Inspector für richtlinienverwaltete Konten:

So deaktivieren Sie die richtlinienverwaltete Amazon Inspector Inspector-Aktivierung

1. Melden Sie sich mit dem AWS Organizations Verwaltungskonto oder dem Policy-Administratorkonto an.
2. Ändern Sie die Organisationsrichtlinie, um die Scantypen in den Regionen, in denen Sie Inspector deaktivieren möchten, explizit auf deaktiviert zu setzen. Sie müssen den Inhalt der Richtlinie aktualisieren, um die deaktivierten Regionen für die Scantypen anzugeben, die Sie deaktivieren möchten.
3. AWS Organizations wendet die Richtlinienänderungen automatisch an und Amazon Inspector deaktiviert die angegebenen Scantypen in den betroffenen Konten.

Detaillierte Anweisungen zum Ändern oder Trennen von Unternehmensrichtlinien finden Sie in der AWS Organizations Dokumentation zu den Richtlinien von Amazon Inspector.

> **Note**
>
> Wenn Sie eine Unternehmensrichtlinie von Konten trennen, behalten diese Konten ihre aktuellen Amazon Inspector Inspector-Einstellungen bei (aktiviert oder deaktiviert, je nach der zuletzt angewendeten Richtlinie). Die Konten werden nicht mehr durch die Richtlinie verwaltet und können dann ihre Amazon Inspector Inspector-Einstellungen unabhängig oder über den delegierten Administrator verwalten.

Amazon Inspector deaktivieren

> **Note**
>
> Bevor Sie Amazon Inspector deaktivieren, sollten Sie erwägen, Ihre Ergebnisse zu exportieren.

Console

Um Amazon Inspector zu deaktivieren

1. Melden Sie sich mit Ihren Anmeldeinformationen an und öffnen Sie dann die Amazon Inspector Inspector-Konsole unter https://console.aws.amazon.com/inspector/v2/home.
2. Wählen Sie mithilfe der AWS-Region Auswahl in der oberen rechten Ecke der Seite die Region aus, in der Sie Amazon Inspector deaktivieren möchten.
3. Wählen Sie im Navigationsbereich Allgemeine Einstellungen aus.
4. Wählen Sie „Inspector deaktivieren“.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie in das Textfeld deaktivieren ein und wählen Sie dann Inspector deaktivieren.
6. (Empfohlen) Wiederholen Sie diese Schritte in jeder Region, für die Sie Amazon Inspector deaktivieren möchten.

API

Führen Sie den Vorgang „API deaktivieren“ aus. Geben Sie in der Anfrage das Konto an, das IDs Sie deaktivieren `resourceTypes` möchten, und `EC2, ECR, LAMBDA` für die Deaktivierung aller Scans, wodurch das Konto deaktiviert wird.

Amazon Inspector Inspector-Kontingente

In diesem Abschnitt sind die Amazon Inspector Inspector-Kontingente pro aufgeführt AWS-Region.

Ressource	Standard	Kommentare
Mitgliedskonten	10.000	Die maximale Anzahl von Mitgliedskonten, die einem delegierten Administratorkonto von Amazon Inspector zugeordnet sind. Das Limit basiert auf den Kontingenten für AWS Organizations.
Unterdrückungsregeln	500	Die maximale Anzahl an gespeicherten Unterdrückungsregeln pro AWS Konto und Region. Sie können keine Kontingenterhöhung beantragen.
Ergebnisse EC2 des Amazon-Netzwerks	10.000	Die maximale Anzahl von EC2 Amazon-Netzwerkergebnissen pro AWS Konto. Sie können keine Kontingenterhöhung beantragen.
CIS-Scan-Konfigurationen	500	Die maximale Anzahl von CIS-Scan-Konfigurationen. Sie können keine

Ressource	Standard	Kommentare
		Kontingenterhöhung beantragen.

Eine Liste der mit Amazon Inspector Classic verknüpften Kontingente finden Sie unter Amazon Inspector Classic-Servicekontingente in der Allgemeine AWS-Referenz. Eine Liste der damit AWS Organizations verbundenen Kontingente finden Sie unter AWS Organizations Servicekontingenten in der Allgemeine AWS-Referenz.

Regionen und Endpunkte

Dieses Thema enthält Tabellen mit Endpunkten für Amazon Inspector und Amazon Inspector Scan. Es enthält auch Tabellen, die zeigen, welche Amazon Inspector Inspector-Funktionen AWS-Regionen unterstützen. Informationen darüber, AWS-Regionen wo Amazon Inspector verfügbar ist, finden Sie unter Amazon Inspector Inspector-Endpunkt und Kontingente in der Allgemeine Amazon Web Services-Referenz.

Service-Endpunkte für Amazon Inspector

Die folgende Tabelle zeigt die Service-Endpunkte für Amazon Inspector. Die Namenskonvention für Amazon Inspector Inspector-Endpunkte lautet`inspector2.`*`Region`*`.amazonaws.com`.

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Ohio)	us-east-2	inspector2.us-east-2.amazonaws.com	HTTPS
		inspector2-fips.us-east-2.amazonaws.com	HTTPS
USA Ost (Nord-Virginia)	us-east-1	inspector2.us-east-1.amazonaws.com	HTTPS
		inspector2-fips.us-east-1.amazonaws.com	HTTPS
USA West (Nordkalifornien)	us-west-1	inspector2.us-west-1.amazonaws.com	HTTPS
		inspector2-fips.us-west-1.amazonaws.com	HTTPS
USA West (Oregon)	us-west-2	inspector2.us-west-2.amazonaws.com	HTTPS
		inspector2-fips.us-west-2.amazonaws.com	HTTPS
Afrika (Kapstadt)	af-south-1	inspector2.af-south-1.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
Asien-Pazifik (Hongkong)	ap-east-1	inspector2.ap-east-1.amazonaws.com	HTTPS	
Asien-Pazifik (Hyderabad)	ap-south-2	inspector2.ap-south-2.amazonaws.com	HTTPS	
Asien-Pazifik (Jakarta)	ap-southeast-3	inspector2.ap-southeast-3.amazonaws.com	HTTPS	
Asien-Pazifik (Malaysia)	ap-southeast-5	inspector2.ap-southeast-5.amazonaws.com	HTTPS	
Asien-Pazifik (Melbourne)	ap-southeast-4	inspector2.ap-southeast-4.amazonaws.com	HTTPS	
Asien-Pazifik (Mumbai)	ap-south-1	inspector2.ap-south-1.amazonaws.com	HTTPS	
Asien-Pazifik (Osaka)	ap-northeast-3	inspector2.ap-northeast-3.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
Asien-Pazifik (Seoul)	ap-northeast-2	inspector2.ap-northeast-2.amazonaws.com	HTTPS	
Asien-Pazifik (Singapur)	ap-southeast-1	inspector2.ap-southeast-1.amazonaws.com	HTTPS	
Asien-Pazifik (Sydney)	ap-southeast-2	inspector2.ap-southeast-2.amazonaws.com	HTTPS	
Asien-Pazifik (Taipeh)	ap-east-2	inspector2.ap-east-2.amazonaws.com	HTTPS	
Asien-Pazifik (Thailand)	ap-southeast-7	inspector2.ap-southeast-7.amazonaws.com	HTTPS	
Asien-Pazifik (Tokio)	ap-northeast-1	inspector2.ap-northeast-1.amazonaws.com	HTTPS	
Kanada (Zentral)	ca-central-1	inspector2.ca-central-1.amazonaws.com	HTTPS	
Kanada West (Calgary)	ca-west-1	inspector2.ca-west-1.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
Europa (Frankfurt)	eu-central-1	inspector2.eu-central-1.amazonaws.com	HTTPS	
Europa (Irland)	eu-west-1	inspector2.eu-west-1.amazonaws.com	HTTPS	
Europa (London)	eu-west-2	inspector2.eu-west-2.amazonaws.com	HTTPS	
Europa (Mailand)	eu-south-1	inspector2.eu-south-1.amazonaws.com	HTTPS	
Europa (Paris)	eu-west-3	inspector2.eu-west-3.amazonaws.com	HTTPS	
Europa (Spanien)	eu-south-2	inspector2.eu-south-2.amazonaws.com	HTTPS	
Europa (Stockholm)	eu-north-1	inspector2.eu-north-1.amazonaws.com	HTTPS	
Europa (Zürich)	eu-central-2	inspector2.eu-central-2.amazonaws.com	HTTPS	
Israel (Tel Aviv)	il-central-1	inspector2.il-central-1.amazonaws.com	HTTPS	
Mexiko (Zentral)	mx-central-1	inspector2.mx-central-1.amazonaws.com	HTTPS	
Naher Osten (Bahrain)	me-south-1	inspector2.me-south-1.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Naher Osten (VAE)	me-central-1	inspector2.me-central-1.amazonaws.com	HTTPS
Südamerika (São Paulo)	sa-east-1	inspector2.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-East)	us-gov-east-1	inspector2.us-gov-east-1.amazonaws.com inspector2-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS
AWS GovCloud (US-West)	us-gov-west-1	inspector2.us-gov-west-1.amazonaws.com inspector2-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS

Endpunkte für die Amazon Inspector Scan API

Die folgende Tabelle zeigt die regionalen Endpunkte, die beim Aufrufen der Amazon Inspector Scan API verwendet werden können. Wenn Sie die API verwenden, müssen Sie den Endpunkt und die entsprechende Region für die AWS Region angeben, in der Sie derzeit authentifiziert sind.

Die Namenskonvention für Amazon Inspector Scan-Endpunkte lautet`inspector-scan.`*`region`*`.amazonaws.com`. Wenn Sie beispielsweise authentifiziert sind, würden Sie den Endpunkt verwenden`us-west-2`, `inspector-scan.us-west-2.amazonaws.com` um die API aufzurufen. `inspector-scan`

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Ohio)	us-east-2	inspector-scan.us-east-2.amazonaws.com inspector-scan-fips.us-east-2.amazonaws.com	HTTPS HTTPS
USA Ost (Nord-Virginia)	us-east-1	inspector-scan.us-east-1.amazonaws.com inspector-scan-fips.us-east-1.amazonaws.com	HTTPS HTTPS
USA West (Nordkalifornien)	us-west-1	inspector-scan.us-west-1.amazonaws.com inspector-scan-fips.us-west-1.amazonaws.com	HTTPS HTTPS
USA West (Oregon)	us-west-2	inspector-scan.us-west-2.amazonaws.com inspector-scan-fips.us-west-2.amazonaws.com	HTTPS HTTPS
Afrika (Kapstadt)	af-south-1	inspector-scan.af-south-1.amazonaws.com	HTTPS
Asien-Pazifik (Hongkong)	ap-east-1	inspector-scan.ap-east-1.amazonaws.com	HTTPS
Asien-Pazifik (Hyderabad)	ap-south-2	inspector-scan.ap-south-2.amazonaws.com	HTTPS
Asien-Pazifik (Jakarta)	ap-southeast-3	inspector-scan.ap-southeast-3.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
Asien-Pazifik (Malaysia)	ap-southeast-5	inspector-scan.ap-southeast-5.amazonaws.com	HTTPS	
Asien-Pazifik (Melbourne)	ap-southeast-4	inspector-scan.ap-southeast-4.amazonaws.com	HTTPS	
Asien-Pazifik (Mumbai)	ap-south-1	inspector-scan.ap-south-1.amazonaws.com	HTTPS	
Asien-Pazifik (Osaka)	ap-northeast-3	inspector-scan.ap-northeast-3.amazonaws.com	HTTPS	
Asien-Pazifik (Seoul)	ap-northeast-2	inspector-scan.ap-northeast-2.amazonaws.com	HTTPS	
Asien-Pazifik (Singapur)	ap-southeast-1	inspector-scan.ap-southeast-1.amazonaws.com	HTTPS	
Asien-Pazifik (Sydney)	ap-southeast-2	inspector-scan.ap-southeast-2.amazonaws.com	HTTPS	
Asien-Pazifik (Taipeh)	ap-east-2	inspector-scan.ap-east-2.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
Asien-Pazifik (Thailand)	ap-southeast-7	inspector-scan.ap-southeast-7.amazonaws.com	HTTPS	
Asien-Pazifik (Tokio)	ap-northeast-1	inspector-scan.ap-northeast-1.amazonaws.com	HTTPS	
Kanada (Zentral)	ca-central-1	inspector-scan.ca-central-1.amazonaws.com	HTTPS	
Kanada West (Calgary)	ca-west-1	inspector-scan.ca-west-1.amazonaws.com	HTTPS	
Europa (Frankfurt)	eu-central-1	inspector-scan.eu-central-1.amazonaws.com	HTTPS	
Europa (Irland)	eu-west-1	inspector-scan.eu-west-1.amazonaws.com	HTTPS	
Europa (London)	eu-west-2	inspector-scan.eu-west-2.amazonaws.com	HTTPS	
Europa (Mailand)	eu-south-1	inspector-scan.eu-south-1.amazonaws.com	HTTPS	
Europa (Paris)	eu-west-3	inspector-scan.eu-west-3.amazonaws.com	HTTPS	
Europa (Spanien)	eu-south-2	inspector-scan.eu-south-2.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Europa (Stockholm)	eu-north-1	inspector-scan.eu-north-1.amazonaws.com	HTTPS
Europa (Zürich)	eu-central-2	inspector-scan.eu-central-2.amazonaws.com	HTTPS
Israel (Tel Aviv)	il-central-1	inspector-scan.il-central-1.amazonaws.com	HTTPS
Mexiko (Zentral)	mx-central-1	inspector-scan.mx-central-1.amazonaws.com	HTTPS
Naher Osten (Bahrain)	me-south-1	inspector-scan.me-south-1.amazonaws.com	HTTPS
Naher Osten (VAE)	me-central-1	inspector-scan.me-central-1.amazonaws.com	HTTPS
Südamerika (São Paulo)	sa-east-1	inspector-scan.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-East)	us-gov-east-1	inspector-scan.us-gov-east-1.amazonaws.com inspector-scan-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS
AWS GovCloud (US-West)	us-gov-west-1	inspector-scan.us-gov-west-1.amazonaws.com inspector-scan-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS

Region-specific Verfügbarkeit von Funktionen

In diesem Abschnitt wird die Verfügbarkeit der Amazon Inspector Inspector-Funktionen von beschrieben AWS-Region.

Agentenloses EC2-Scannen für Amazon EC2 EC2-Regionen

Die folgende Tabelle zeigt, AWS-Regionen wo agentenloses Scannen für Amazon EC2 derzeit verfügbar ist.

Name der Region	Regionscode
USA Ost (Nord-Virginia)	us-east-1
USA Ost (Ohio)	us-east-2
USA West (Nordkalifornien)	us-west-1
USA West (Oregon)	us-west-2
Afrika (Kapstadt)	af-south-1
Asien-Pazifik (Hongkong)	ap-east-1
Asien-Pazifik (Tokio)	ap-northeast-1
Asien-Pazifik (Seoul)	ap-northeast-2
Asia Pacific (Osaka)	ap-northeast-3
Asien-Pazifik (Mumbai)	ap-south-1
Asien-Pazifik (Hyderabad)	ap-south-2
Asien-Pazifik (Singapur)	ap-southeast-1
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Jakarta)	ap-southeast-3
Asien-Pazifik (Melbourne)	ap-southeast-4

Name der Region	Regionscode
Asien-Pazifik (Malaysia)	ap-southeast-5
Asien-Pazifik (Thailand)	ap-southeast-7
Kanada (Zentral)	ca-central-1
Kanada West (Calgary)	ca-west-1
Europa (Stockholm)	eu-north-1
Europa (Frankfurt)	eu-central-1
Europa (Zürich)	eu-central-2
Europa (Irland)	eu-west-1
Europa (London)	eu-west-2
Europa (Paris)	eu-west-3
Europa (Milan)	eu-south-1
Europa (Spain)	eu-south-2
Israel (Tel Aviv)	il-central-1
Naher Osten (VAE)	me-central-1
Naher Osten (Bahrain)	me-south-1
Mexiko (Zentral)	mx-central-1
Südamerika (São Paulo)	sa-east-1
AWS GovCloud (US-East)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1

Lambda-Code-Scanning-Regionen

Die folgende Tabelle zeigt, AWS-Regionen wo Lambda-Code-Scanning derzeit verfügbar ist.

Name der Region	Regionscode
USA Ost (Nord-Virginia)	us-east-1
USA West (Oregon)	us-west-2
USA Ost (Ohio)	us-east-2
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1
Europa (Frankfurt)	eu-central-1
Europa (Irland)	eu-west-1
Europa (London)	eu-west-2
Europa (Stockholm)	eu-north-1
Asien-Pazifik (Singapur)	ap-southeast-1

Important

Wenn Sie versuchen, Lambda-Code-Scanning mit der Amazon Inspector Enable API in einer AWS-Region Umgebung zu aktivieren, in der Lambda-Code-Scans nicht verfügbar sind, wird Ihnen die folgende Fehlermeldung „Zugriff verweigert“ angezeigt:

```
An error occurred (AccessDeniedException) when calling the Enable operation:
 Lambda code scanning is not supported in unsupported-AWS-Region
```

Amazon Inspector Code — Sicherheitsregionen

Die folgende Tabelle zeigt AWS-Regionen , wo Amazon Inspector Code Security derzeit verfügbar ist.

Name der Region	Regionscode
USA Ost (Nord-Virginia)	us-east-1
USA West (Oregon)	us-west-2
USA Ost (Ohio)	us-east-2
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1
Europa (Frankfurt)	eu-central-1
Europa (Irland)	eu-west-1
Europa (London)	eu-west-2
Europa (Stockholm)	eu-north-1
Asien-Pazifik (Singapur)	ap-southeast-1

AWS GovCloud (US) Regionen

Die neuesten Informationen finden Sie unter Amazon Inspector im AWS GovCloud (US) Benutzerhandbuch.

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen des Amazon Inspector Inspector-Benutzerhandbuchs ab November 2021 beschrieben. Um Benachrichtigungen über Aktualisierungen der Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Amazon Inspector Produktupdates

Änderung	Beschreibung	Datum
Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator möglicher weise eine Schwachst ellensuche für generiert. CVE-2026-85747 Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von dieser Sicherheitsanfälli gkeit nicht betroffen ist. Dieses Problem wird behoben, indem der Amazon Inspector SBOM Generator auf Version 1.17.2 oder höher aktualisiert wird.	19. Oktober 2026
Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator möglicher weise eine Schwachst ellensuche für generiert. CVE-2026-85747 Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von dieser Sicherheitsanfälli	16. Oktober 2026

	gkeit nicht betroffen ist. Dieses Problem wird in der nächsten Version des Amazon Inspector SBOM Generators behoben. In der Zwischenzeit können Kunden diese Sicherhei tsanfälligkeit getrost ignorieren oder unterdrücken.	
Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator Sicherhei tslücken für CVE-2026-56866 CVE-2026-78659, CVE-2026-78660, CVE-2026-78663,, CVE-2026-78667, CVE-2026-78669, CVE-2026-94439, CVE-2026-94440, CVE-2026-94448 CVE-2026-97030 CVE-2026-97031, und generiert. CVE-2026-97032 Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von diesen Sicherheitslücken nicht betroffen ist. Diese Probleme werden behoben, indem der Amazon Inspector SBOM Generator auf Version 1.17.1 oder höher aktualisiert wird.	9. Oktober 2026

Version 1.17.0 von Amazon Inspector SBOM Generator	Amazon Inspector SBOM Generator 1.17.0 veröffent licht. Download-Links und Versionsdetails finden Sie unter Frühere Versionen des Amazon Inspector SBOM Generators.	6. Oktober 2026
Amazon Inspector fügt maschinelles Scannen von Bildern auf Abruf hinzu	Sie können jetzt bei Bedarf ein einzelnes Maschinen abbild (AMI) scannen, ohne Amazon Inspector zu aktiviere n. Starten Sie einen Scan mit dem StartOnDemandScan API-Vorgang oder von der Amazon EC2 EC2-Konso le aus, verfolgen Sie ihn mit GetOnDemandScanStatus und und ListOnDemandScans erhalten Sie über Amazon EventBridge Benachric htigungen zu Ergebnissen und Abschluss des Scans. Weitere Informationen finden Sie unter Ausführen von On-Demand-Computerbildscans mit Amazon Inspector.	17. September 2026

Version 1.16.0 von Amazon Inspector SBOM Generator	Der Amazon Inspector SBOM Generator 1.16.0 bietet Unterstützung für die Erkennung von Metabase Installationen undMozilla Firefox, und Thunderbird Installationen auf Firefox ESRLinux, und. Windows macOS Diese Version bietet außerdem die Identifizierung vonRed Hat Hardened Images, gibt bei Container -Scans eine `amazon:inspector:sbom_generator:image:base_layer` Eigenschaft für die Bild-Metadaten-Komponente aus und fügt die `sbomgen.get_home_dirs()` Funktion der Plugin-API hinzu. Lua Weitere Informationen finden Sie unter Frühere Versionen des Amazon Inspector SBOM Generators.	8. September 2026

Updates für das Amazon Inspector SSM-Plugin, den Amazon Inspector SBOM Generator und den Amazon Inspector VM Scanner	Amazon Inspector ist sich eines Szenarios bewusst, für das das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner Sicherheitslücken ermitteln können. CVE-2026-56854 Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner von diesen Sicherhei tslücken nicht betroffen sind. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plug-ins, des Amazon Inspector SBOM Generators und des Amazon Inspector VM Scanners behoben.	31. August 2026
Aktualisierte unterstützte CIS-Benchmark-Versionen	Amazon Inspector hat die unterstützten Benchmarks für Rocky 8/9, RHEL 8, Ubuntu 22 aktualisiert. 04/24.04 und Windows Server/. 2016/2019 2022/2025 Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector	24. August 2026

Updates für das Amazon Inspector SSM-Plugin, den Amazon Inspector SBOM Generator und den Amazon Inspector VM Scanner	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner Sicherheitslücken für und generieren können. CVE-2026-61711 CVE-2026-61712 Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner von diesen Sicherheitslücken nicht betroffen sind. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plug-ins, des Amazon Inspector SBOM Generators und des Amazon Inspector VM Scanners behoben.	19. August 2026

Updates für das Amazon Inspector SSM-Plugin, den Amazon Inspector SBOM Generator und den Amazon Inspector VM Scanner	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plug-In, Amazon Inspector SBOM Generator und Amazon Inspector VM Scanner Sicherheitslücken für CVE-2026-39821, CVE-2026-46600, CVE-2026-33818,, CVE-2026-56859, CVE-2026-56858 CVE-2026-56864 CVE-2026-56853 CVE-2026-56860, CVE-2026-56865 und generieren können. CVE-2026-56862 Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner von diesen Sicherheitslücken nicht betroffen sind. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plug-ins, des Amazon Inspector SBOM Generators und des Amazon Inspector VM Scanners behoben.	14. August 2026

Aktualisierte Anleitung für Amazon EC2-Scans	Klarere Leitlinien für Amazon EC2-Scans zu verschiedenen Themen. Wir empfehlen jetzt Enhanced EC2 Scanning mit dem Amazon Inspector VM Scanner gegenüber dem veralteten Amazon Inspector SSM-Plugin und stellen fest, dass es zu konsisten teren Ergebnissen führt — insbesondere bei Windows Es wurde dokumentiert, dass benutzerdefinierte Pfade zur Tiefeninspektion auf LinuxWindows, und macOS wenn Sie Enhanced EC2 Scanning verwenden , unterstützt werden, und es wurden die standardm äßigen Scan-Speicherorte für jedes Betriebssystem geklärt. Es wurde eine Anleitung hinzugefügt, dass Änderunge n an den Dateien der Amazon Inspector VM `systemd` Scanner-Einheit bei Paket-Updates verloren gehen, und es wurde dokumentiert, wie persistente Überschreibungen vorgenommen werden können. Weitere Informati onen finden Sie unter Amazon Inspector VM Scanner und Benutzerdefinierte Pfade für Amazon Inspector Deep Inspection.	28. Juli 2026

Updates für das Amazon Inspector SSM-Plugin, den Amazon Inspector SBOM Generator und den Amazon Inspector VM Scanner	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner Sicherheitslücken für und generieren können. CVE-2026-46600 CVE-2026-56852 Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner von diesen Sicherheitslücken nicht betroffen sind. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plug-ins, des Amazon Inspector SBOM Generators und des Amazon Inspector VM Scanners behoben.	24. Juli 2026

Updates für das Amazon Inspector SSM-Plugin, den Amazon Inspector SBOM Generator und den Amazon Inspector VM Scanner	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner Sicherheitslücken für und generieren können. CVE-2026-39822 CVE-2026-42505 Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner von diesen Sicherheitslücken nicht betroffen sind. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plug-ins, des Amazon Inspector SBOM Generators und des Amazon Inspector VM Scanners behoben.	9. Juli 2026
Aktualisierte Funktionalität	Amazon Inspector unterstützt jetzt die Einstellungen für die Dauer des erneuten Scans von Amazon ECR für 3 Tage und 7 Tage sowohl für die Dauer des erneuten Scans von Bildern als auch für die Dauer des Bild-Push-Datums. Weitere Informationen finden Sie unter Konfiguration der Dauer des Amazon ECR-Neuscans.	9. Juli 2026

Neuer Inhalt	Inhalt hinzugefügt, der alle Programmiersprachen auflistet, die der Amazon Inspector VM Scanner derzeit unterstützt. Weitere Informationen finden Sie unter Unterstützte Programmiersprachen: Amazon Inspector VM Scanner.	16. Juni 2026
Updates für das Amazon Inspector SSM-Plugin, den Amazon Inspector SBOM Generator und den Amazon Inspector VM Scanner	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner Sicherheitslücken für CVE-2026-27145, CVE-2026-42504 und generieren können. CVE-2026-42507 Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin, der Amazon Inspector SBOM Generator und der Amazon Inspector VM Scanner von diesen Sicherheitslücken nicht betroffen sind. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plug-ins, des Amazon Inspector SBOM Generators und des Amazon Inspector VM Scanners behoben.	3. Juni 2026

Neues Feature	Um erweiterte Schwachstellenanalysen von Amazon Elastic Compute Cloud-Instances durchzuführen, können Sie jetzt den Amazon Inspector VM Scanner anstelle des Amazon Inspector SSM-Plug-ins verwenden. Der Amazon Inspector VM Scanner bietet mehrere Vorteile, wie z. B. eine detailliertere Paketerfassung, und er verbraucht weniger Rechenressourcen. Weitere Informationen finden Sie unter Amazon Inspector VM Scanner.	29. Mai 2026
Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator Sicherheitslücken für CVE-2026-39821 CVE-2026-39827, und generieren könnte. CVE-2026-39824 Es wurde bestätigt, dass Amazon Inspector von diesen Sicherheitslücken nicht betroffen ist. Diese Probleme werden in der nächsten Version des Amazon Inspector SBOM Generators behoben.	28. Mai 2026

Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, für das der Amazon Inspector SBOM Generator Sicherhei tslücken ermitteln könnte. CVE-2026-29181 Es wurde bestätigt, dass der Amazon Inspector SBOM Generator nicht von dieser Sicherhei tsanfälligkeit betroffen ist. Dieses Problem kann behoben werden, indem Sie den Amazon Inspector SBOM Generator auf Version 1.12.2 oder höher aktualisieren.	27. April 2026
Updates für das Amazon Inspector SSM-Plugin	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin Sicherhei tslücken für CVE-2026-32283, CVE-2026-33810, CVE-2026-32289 CVE-2026-32281 CVE-2026-32288, und CVE-2026-32282 generiere n könnte. Es wurde bestätigt , dass das Amazon Inspector SSM-Plugin von diesen Sicherheitslücken nicht betroffen ist. Diese Probleme werden in der nächsten Version des Amazon Inspector SSM-Plugins behoben.	13. April 2026

Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, für das der Amazon Inspector SBOM Generator Sicherhei tslücken ermitteln könnte. CVE-2026-32280 Dieses Problem wurde durch ein Upgrade auf Amazon Inspector SBOM Generator Version 1.12.1 behoben.	9. April 2026
Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator Sicherhei tslücken für CVE-2026-33997 CVE-2026-34040, CVE-2026-33747 und generieren könnte. CVE-2026-33748 Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von diesen Sicherheitslücken nicht betroffen ist. Diese Probleme werden in Amazon Inspector SBOM Generator Version 1.12.0 behoben.	26. März 2026

Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator Sicherhei tslücken für CVE-2026-25679 CVE-2026-27142, und generieren könnte. CVE-2026-27139 Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von diesen Sicherheitslücken nicht betroffen ist. Diese Sicherhei tsanfälligkeit kann durch ein Upgrade der Amazon Inspector SBOM Generator-Version auf 1.11.2 oder höher behoben werden.	11. März 2026
Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, für das der Amazon Inspector SBOM Generator Sicherhei tslücken ermitteln könnte. `CVE-2025-15558` Es wurde bestätigt, dass der Amazon Inspector SBOM Generator davon nicht betroffen ist. CVE-2025-15558 Diese Sicherheitsanfälligkeit kann durch ein Upgrade der Amazon Inspector SBOM Generator-Version auf 1.11.1 oder höher behoben werden.	5. März 2026

Updates für den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, für das der Amazon Inspector SBOM Generator Sicherhei tslücken ermitteln könnte. `CVE-2025-68121` Es wurde bestätigt, dass der Amazon Inspector SBOM Generator davon nicht betroffen ist. CVE-2025-68121 Diese Sicherheitsanfälligkeit kann durch ein Upgrade der Amazon Inspector SBOM Generator-Version auf 1.11.0 oder höher behoben werden.	2. März 2026
Neue verwaltete Richtlinie	Amazon Inspector hat eine neue verwaltete Richtlinie veröffentlicht`AmazonIns pector2ManagedTele metryPolicy` , die Berechtigungen für Telemetri eoperationen von Amazon Inspector gewährt und es dem Service ermöglicht, Paketbest andsdaten für Sicherhei tslücken zu sammeln und zu übertragen. Weitere Informati onen finden Sie unter Aktualisi erungen AWS verwalteter Richtlinien durch Amazon Inspector.	5. Februar 2026

Aktualisierte Richtlinie	Amazon Inspector fügt der mit dem Service verknüpften Rolle namens `AmazonInspector2ServiceRolePolicy` neue Berechtigungen hinzu. Amazon Inspector hat eine neue Berechtigung hinzugefügt, die es Amazon Inspector ermöglicht, Firewall-Metadaten für die Analyse der Netzwerkerreichbarkeit zu beschreiben. Darüber hinaus hat Amazon Inspector zusätzlichen Ressourcenbereich hinzugefügt, damit Amazon Inspector SSM-Verknüpfungen mit SSM-Dokumenten erstellen, aktualisieren und starten kann. `AWS-ConfigureAWSPackage` Weitere Informationen finden Sie unter Service-linked Rollenberechtigungen für Amazon Inspector.	3. Februar 2026

Updates für das Amazon Inspector SSM-Plugin und den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin und der Amazon Inspector SBOM Generator Sicherheitslücken ermitteln können. `CVE-2025-61728`, `CVE-2025-61730`, and `CVE-2025-61726` Diese Sicherheitsanfälligkeiten können durch ein Upgrade der Amazon Inspector SSM-Plug-in-Version auf 1.0.2327.0 oder Amazon Inspector SBOM Generator 1.10.1 oder höher behoben werden.	29. Januar 2026
Updates für das Amazon Inspector SSM-Plugin und den Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem das Amazon Inspector SSM-Plugin und der Amazon Inspector SBOM Generator Sicherheitslücken ermitteln könnten. `CVE-2025-61729` Es wurde bestätigt, dass diese Anwendungen von diesem CVE nicht betroffen sind. Wir arbeiten derzeit an Verbesserungen, um dieses Problem zu beheben. In der Zwischenzeit können Kunden diese Sicherheitsanfälligkeit getrost ignorieren oder unterdrücken.	3. Dezember 2025

Aktualisierungen für Amazon Inspector SBOM Generator	Amazon Inspector ist sich eines Szenarios bewusst, in dem der Amazon Inspector SBOM Generator Sicherheitslücken für `CVE-2025-47914` und generieren könnte. `CVE-2025-58181` Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von diesen CVEs nicht betroffen ist. Wir arbeiten derzeit an Verbesserungen, um diese Erkennungen zu beheben. In der Zwischenzeit können Kunden diese Sicherheitslücken getrost ignorieren oder unterdrücken.	20. November 2025
Neues Feature	Amazon Inspector unterstützt jetzt AWS Organizations Richtlinien für die zentrale Aktivierung und Steuerung aller Unternehmenskonten. Mithilfe der Unternehmensrichtlinien können Sie Amazon Inspector-Scantypen in Ihrem Unternehmen automatisch aktivieren und unbefugte Änderungen verhindern. Weitere Informationen finden Sie im Tutorial Erste Schritte und Mehrere Konten verwalten.	19. November 2025

Aktualisierungen für Amazon Inspector SBOM Generator	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, für das der Amazon Inspector SBOM Generator Sicherheitslücken ermitteln könnte. `CVE-2025-47913` Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von diesem CVE nicht betroffen ist, und es wurde ein Update bereitgestellt, um diese Erkennung zu beheben.	14. November 2025
Aktualisierte Richtlinien	Amazon Inspector fügt den verwalteten Richtlinien neue Berechtigungen hinzu `AmazonInspector2FullAccess_v2` und `AmazonInspector2ReadOnlyAccess`. Die Berechtigungen ermöglichen die Anzeige von Amazon Inspector Inspector-Organisationsrichtlinien und Delegierungskonfigurationen, die durch AWS Organizations Richtlinien festgelegt wurden. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinien für Amazon Inspector.	14. November 2025

Aktualisierungen für Amazon Inspector SBOM Generator	Amazon Inspector aktualisiert die Version von Amazon Inspector SBOM Generator. Weitere Informationen finden Sie unter Frühere Versionen des Amazon Inspector SBOM Generators.	11. November 2025
Aktualisierte Richtlinie	Amazon Inspector fügt der mit dem Service verknüpften Rolle namens `AmazonInspector2ServiceRolePolicy` neue Berechtigungen hinzu. Die Berechtigungen ermöglichen es der Amazon AWS Organizations Inspector-Richtlinie, die Aktivierung und Deaktivierung von Amazon Inspector durchzusetzen. Weitere Informationen finden Sie unter Service-linked Rollenberechtigungen für Amazon Inspector.	10. November 2025

Aktualisierungen für Amazon Inspector SBOM Generator	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, in dem der Amazon Inspector SBOM Generator Sicherheitslücken für `CVE-2025-58188` und generieren könnte. `CVE-2025-61725` Es wurde bestätigt, dass der Amazon Inspector SBOM Generator von diesen CVEs nicht betroffen ist, und Amazon Inspector aktualisiert die Version von Amazon Inspector SBOM Generator. Weitere Informationen finden Sie unter Frühere Versionen des Amazon Inspector SBOM Generators.	4. November 2025
Update für das Plugin	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, in dem das Amazon Inspector SSM-Plugin Sicherheitslücken für `CVE-2025-58188` und `CVE-2025-61725` generieren könnte. Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin von diesen CVEs nicht betroffen ist, und es wurde ein Update bereitgestellt, um diese Erkennung zu beheben.	3. November 2025

Update für das Plugin	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, in dem das Amazon Inspector SSM-Plugin möglicherweise eine Schwachstellensuche für `CVE-2025-47907` generiert. Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin von diesen CVEs nicht betroffen ist, und es wurde ein Update bereitgestellt, um diese Erkennung zu beheben.	8. August 2025
Neue Richtlinie	Amazon Inspector fügt eine neue verwaltete Richtlinie hinzu, die vollen Zugriff auf Amazon Inspector und Zugriff auf andere verwandte Services bietet. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinien für Amazon Inspector.	3. Juli 2025
Aktualisierte Funktionalität	Amazon Inspector ist jetzt in neuer Version erhältlich AWS-Regionen. Weitere Informationen finden Sie unter -Regionen und Endpunkte.	1. Juli 2025

Aktualisierte Funktionalität	Amazon Inspector aktualisiert seine Aufbewahrungsfrist für abgeschlossene Ergebnisse. Amazon Inspector entfernt Ergebnisse nach 3 Tagen, wenn zugehörige Ressourcen gelöscht oder beendet wurden oder nicht mehr gescannt werden können. Weitere Informationen finden Sie unter Grundlegendes zu den Ergebnissen von Amazon Inspector.	25. Juni 2025
Aktualisierte Funktionalität	Amazon Inspector aktualisiert seine unterstützten Betriebssysteme für Amazon EC2-Scanning und Amazon ECR-Scannen. Amazon EC2-Scanning unterstützt jetzt Fedora Version 42 und Ubuntu Version 25.04. Amazon ECR Scanning unterstützt jetzt Alpine Version 3.22, Fedora Version 42 und Ubuntu Version 25.04. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	18. Juni 2025

Neues Feature	Amazon Inspector scannt jetzt den Quellcode von Erstanbieteranwendungen, Abhängigkeiten von Drittanbieteranwendungen und Infrastructure as Code auf Sicherheitslücken. Weitere Informationen finden Sie unter Amazon Inspector Code Security.	17. Juni 2025
Update für das Plugin	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, in dem das Amazon Inspector SSM-Plugin möglicherweise eine Sicherheitslücke für `CVE-2025-0913` und `CVE-2025-4673` entdeckt. Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin von diesen CVEs nicht betroffen ist, und es wurde ein Update bereitgestellt, um diese Erkennung zu beheben.	13. Juni 2025
Neues Feature	Amazon Inspector kann jetzt aktiv verwendete Container-Images und den Zeitpunkt der letzten Verwendung von Container-Images in einem Cluster anzeigen. Weitere Informationen finden Sie unter Zuordnen von Container-Images zu laufenden Containern.	16. Mai 2025

Updates für unterstützte Betriebssysteme	Amazon Inspector fügt Unterstützung hinzu BusyBox für Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	13. Mai 2025
Aktualisierte Richtlinie	Amazon Inspector fügt der mit dem Service verknüpften Rolle namens `AmazonInspector2ServiceRolePolicy` eine neue Berechtigung hinzu. Mit dieser Berechtigung können Sie IP-Adressen und Internet-Gateways beschreiben. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinien für Amazon Inspector.	29. April 2025
Update für das Plugin	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, in dem das Amazon Inspector SSM-Plugin möglicherweise eine Schwachstellensuche für `CVE-2025-22871` generiert. Es wurde bestätigt, dass das Amazon Inspector SSM-Plugin von diesen CVEs nicht betroffen ist, und es wurde ein Update bereitgestellt, um diese Erkennung zu beheben.	21. April 2025

Update für das Plugin	Amazon Inspector wird auf ein Szenario aufmerksam gemacht, in dem das Amazon Inspector SSM-Plugin eine Sicherheitslücke für `CVE-2020-8911` `CVE-2020-8912` , und `CVE-2024-45337` finden könnte. Es wurde bestätigt, dass Amazon Inspector von diesen CVEs nicht betroffen ist, und es wurde ein Update bereitgestellt, um diese Erkennung zu beheben.	18. April 2025
Aktualisierungen des Kapitels Amazon Inspector SBOM Generator	Amazon Inspector aktualisiert die Version von Amazon Inspector SBOM Generator. Weitere Informationen finden Sie unter Frühere Versionen des Amazon Inspector SBOM Generators.	16. April 2025
Aktualisierungen des Kapitels Amazon Inspector SBOM Generator	Amazon Inspector fügt dem Kapitel Amazon Inspector SBOM Generator ein neues Thema hinzu. In diesem Thema wird beschrieben, wie Lizenzinformationen in einer Softwareliste Sbomgen nachverfolgt werden. Weitere Informationen finden Sie unter Lizenzsammlung für Amazon Inspector SBOM Generator.	16. April 2025

Aktualisierungen der verwaltet en Richtlinien	Amazon Inspector fügt Berechtigungen hinzu, die nur Lesezugriff auf Amazon ECS- und Amazon EKS-Aktionen ermöglichen. Weitere Informati onen finden Sie unter Service-linked Rollenberechtigungen für Amazon Inspector.	25. März 2025
Updates für unterstützte Betriebssysteme	Amazon Inspector unterstüt zt das SUSE Linux Enterpris e Server 12.5 Scannen für Amazon EC2 und Amazon ECR nicht mehr. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	21. März 2025
Updates für unterstützte Betriebssysteme	Amazon Inspector fügt Unterstützung für Chainguar d und Wolfi zum Amazon ECR-Scannen hinzu. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	21. März 2025
Aktualisierungen des Inhaltsve rzeichnisses	Amazon Inspector fügt ein Kapitel über das Taggen von Amazon Inspector Inspector -Ressourcen hinzu. Weitere Informationen finden Sie unter Ressourcen zum Taggen von Amazon Inspector.	25. Februar 2025

Aktualisierungen des Inhaltsverzeichnisses	Amazon Inspector fügt dem Kapitel Amazon Inspector SBOM Generator ein neues Thema hinzu. Weitere Informationen finden Sie in der umfassenden Betriebssystemsammlung von Amazon Inspector SBOM Generator.	28. Januar 2025
Aktualisierte Funktionalität	Amazon Inspector erweitert nodejs202.x seine Liste der unterstützten Laufzeiten für Lambda-Standardscans um und. python3.13 Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	24. Januar 2025
Aktualisierte Funktionalität	Amazon Inspector entfernt Oracle Linux (Oracle) 7 und SUSE Linux Enterprise Server (SLES) 15.5 aus seiner Liste der unterstützten Betriebssysteme für Amazon EC2 und Amazon ECR. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	31. Dezember 2024

Aktualisierte Funktionalität	Amazon Inspector erweitert seine Liste der unterstützten Betriebssysteme für Amazon EC2 und Amazon ECR um Ubuntu 24.10. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	12. Dezember 2024
Aktualisierungen des Inhaltsverzeichnisses	Amazon Inspector fügt dem Kapitel Amazon Inspector SBOM Generator neue Themen hinzu. Weitere Informationen finden Sie unter Amazon Inspector SBOM Generator.	9. Dezember 2024
Aktualisierte Funktionalität	Amazon Inspector aktualisiert die `amazon:inspector:sbom_generator` Tabelle, um Namespaces hinzuzufügen und zu entfernen. Weitere Informationen finden Sie unter Verwenden von CyclonedX-Namespaces mit Amazon Inspector.	9. Dezember 2024
Aktualisierte Funktionalität	Amazon Inspector aktualisiert seine CI/CD Integrationsfunktion, um Scanaktionen mit zu unterstützen CodePipeline. Weitere Informationen finden Sie unter Amazon Inspector Scan-Aktionen verwenden mit CodePipeline.	26. November 2024

Aktualisierungen des Inhaltsverzeichnisses	Amazon Inspector organisiert das Inhaltsverzeichnis neu und enthält nun ein Kapitel für den Amazon Inspector SBOM Generator. Weitere Informationen finden Sie unter Amazon Inspector SBOM Generator.	22. November 2024
Aktualisierte Funktionalität	Amazon Inspector entfernt Fedora 39 von seiner Liste der unterstützten Betriebssysteme für Amazon EC2 und Amazon ECR. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	22. November 2024
Aktualisierte Funktionalität	Amazon Inspector entfernt Alpine 3.17 aus seiner Liste der unterstützten Betriebssysteme für Amazon ECR. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	22. November 2024
Aktualisierte Funktionalität	Amazon Inspector fügt Sbomgen Versionen zu früheren Versionen des Amazon Inspector SBOM Generators hinzu.	19. November 2024

Aktualisierte Funktionalität	Amazon Inspector fügt AL2 als unterstützte Laufzeit hinzu. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Programmiersprachen für Amazon Inspector.	26. August 2024
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Erklärung hinzugefügt. Die neue Anweisung ermöglicht es Amazon Inspector, Funktions-Tags zurückzugeben AWS Lambda.	31. Juli 2024
Aktualisierte Funktionalität	Amazon Inspector veröffentlicht neue Sicherheitskontrollen. Weitere Informationen finden Sie unter Amazon Inspector-Steuerelemente im AWS Security Hub CSPM Benutzerhandbuch.	11. Juli 2024
Aktualisierte Funktionalität	Der Amazon Inspector SBOM Generator scannt jetzt Dockerfiles und Docker-Container-Images auf Fehlkonfigurationen, die zu Sicherheitslücken führen können. Weitere Informationen finden Sie unter Amazon Inspector Dockerfile-Checks.	10. Juni 2024

Aktualisierte Funktionalität	Amazon Inspector aktualisiert seine CI/CD Integrationsfunktion, um CodeCatalyst Aktionen zu unterstützen, sodass Sie Amazon Inspector Inspector-Schwachstellenscans zu Ihren CodeCatalyst Workflows hinzufügen können. Weitere Informationen finden Sie unter CodeCatalyst Aktionen verwenden.	7. Juni 2024
Aktualisierte Funktionalität	Amazon Inspector bietet eine Option zum Herunterladen einer CSV-Datei mit CIS-Scanergebnissen. Weitere Informationen finden Sie unter CIS-Scanergebnisse anzeigen und herunterladen in Center for Internet Security (CIS) - Scans für Amazon EC2 EC2-Instances.	3. Mai 2024
Aktualisierte Funktionalität	Amazon Inspector aktualisiert seine CI/CD Integrationsfunktion zur UnterstützungGitHub Actions, sodass Sie Amazon Inspector Inspector-Schwachstellenscans zu Ihren GitHub Workflows hinzufügen können. Weitere Informationen finden Sie unter Amazon Inspector verwenden mit GitHub Actions.	29. April 2024

Aktualisierte Funktionalität	Amazon Inspector aktualisiert die verwaltete Richtlinie `AmazonInspector2FullAccess`, sodass die serviceverknüpfte Rolle `AWSServiceRoleForAmazonInspector2Agentless` erstellt wird. Auf diese Weise können Benutzer agentenbasiertes Scannen und agentenloses Scannen durchführen, wenn sie Amazon Inspector aktivieren.	24. April 2024
Aktualisierte Funktionalität	Amazon Inspector aktualisiert die Aufbewahrungsfrist für abgeschlossene Ergebnisse von 30 Tagen auf 7 Tage. Weitere Informationen finden Sie unter Grundlegendes zu den Ergebnissen in Amazon Inspector.	12. Februar 2024
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Erklärung hinzugefügt. Die neue Anweisung ermöglicht es Amazon Inspector, CIS-Scans für Ihre Instance zu starten.	23. Januar 2024

Neue Richtlinie	Amazon Inspector hat eine neue Richtlinie, AmazonInspector2ManagedCisPolicyPolicy, hinzugefügt, die Sie als Teil eines Instance-Profils verwenden können, um CIS-Scans auf einer Instance zuzulassen.	23. Januar 2024
Neue Funktion	Amazon Inspector aktualisiert jetzt die Dauer des ECR-Rescans von Container-Images, wenn Sie sie abrufen. Informationen zum Ändern der Dauer des erneuten Scans auf der Grundlage von Push- oder Pull-Daten finden Sie unter Konfiguration der ECR-Rescan-Dauer.	23. Januar 2024
Neue Funktion	Amazon Inspector kann jetzt Center for Internet Security (CIS) -Scans auf EC2-Instances ausführen. Weitere Informationen finden Sie unter Amazon Inspector CIS-Scans.	23. Januar 2024
Neue Funktion	Amazon Inspector kann jetzt Container-Bilder in Ihren CI/CD Pipelines scannen. Weitere Informationen finden Sie unter CI/CD Integration mit Amazon Inspector.	30. November 2023

Neue Richtlinie	Amazon Inspector hat eine neue Richtlinie hinzugefügt, die es Amazon Inspector ermöglicht, Amazon EBS-Snapshots von Ihrer EC2-Instance für agentenloses Scannen zu scannen. Weitere Informationen zu dieser Richtlinie finden Sie unter Agentloses Scannen.	27. November 2023
Neue Funktion	Amazon Inspector unterstützt jetzt das Scannen unterstützter Linux-Amazon-EC2-Instances ohne SSM-Agenten durch agentenloses Scannen. Weitere Informationen finden Sie unter Agentloses Scannen.	27. November 2023
Neue unterstützte Ressourcen	Amazon Inspector unterstützt jetzt das Scannen von macOS Amazon EC2 EC2-Instances. Siehe Unterstützte Betriebssysteme: Amazon EC2 scannt nach unterstützten macOS-Versionen.	05. Oktober 2023
Neue Regionen	Amazon Inspector ist jetzt in Asien-Pazifik (Jakarta), Afrika (Kapstadt), Asien-Pazifik (Osaka) und Europa (Zürich) verfügbar.	29. September 2023
Neues Feature	Sie können jetzt EC2-Instances mithilfe von Ausschluss-Tags von Amazon Inspector-Scans ausschließen.	14. September 2023

Neues Feature	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, Netzwerkkonfigurationen von Amazon EC2 EC2-Instances zu scannen, die Teil der Elastic Load Balancing Balancing-Zielgruppen sind.	31. August 2023
Neues Feature	Amazon Inspector bietet jetzt Informationen zu Sicherheitslücken für gefundene Sicherheitslücken in Paketen.	31. Juli 2023
Aktualisierte Funktionalität	Amazon Inspector hat neue Berechtigungen hinzugefügt, die es Benutzern mit Lesezugriff ermöglichen, Software Bill of Materials (SBOM) für ihre Ressourcen zu exportieren.	29. Juni 2023
Neues Feature	Sie können jetzt SBOM für Ressourcen exportieren, die von Amazon Inspector gescannt werden.	13. Juni 2023

Neues Feature	Das Scannen von Lambda-Code ist jetzt allgemein verfügbar. Es wurden neue Funktionen hinzugefügt, mit denen Sie Code verschlüsseln können, der in Ihren Ergebnissen beim Lambda-Code-Scannen identifiziert wurde. Darüber hinaus bietet das Lambda-Code-Scannen jetzt Vorschläge zur Behebung von Neuschreibungen Ihres Codes.	13. Juni 2023
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ReadOnlyAccessRichtlinie eine neue Erklärung hinzugefügt. Die neuen Anweisungen ermöglichen es Benutzern, die nur lesen können, Details zum Status und zu den Ergebnissen des Lambda-Code-Scans für ihr Konto abzurufen.	2. Mai 2023
Neues Feature	Amazon Inspector hat die Suche nach Sicherheitslücken in der Datenbank hinzugefügt, mit der Sie überprüfen können, ob Amazon Inspector ein bestimmtes CVE abdeckt.	1. Mai 2023

Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie neue Berechtigungen hinzugefügt, die es Amazon Inspector ermöglichen, AWS CloudTrail serviceverknüpfte Kanäle in Ihrem Konto zu erstellen, wenn Sie Lambda-Scanning aktivieren. Dadurch kann Amazon Inspector CloudTrail Ereignisse in Ihrem Konto überwachen.	30. April 2023
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2FullAccessRichtlinie eine neue Erklärung hinzugefügt. Die neue Erklärung ermöglicht es Benutzern, Details zu den beim Lambda-Code-Scannen gefundenen Sicherheitslücken abzurufen.	17. April 2023
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Erklärung hinzugefügt. Die neue Erklärung ermöglicht es Amazon Inspector, Informationen über die benutzerdefinierten Pfade, die Sie für Amazon EC2 Deep Inspection definiert haben, an Amazon EC2 Systems Manager zu senden.	17. April 2023

Neues Feature	Amazon Inspector bietet zusätzliche Unterstützung für Linux EC2-Instances in Form von Amazon Inspector Deep Inspection, die Ihre Instances auf Paketschwachstellen in Programmiersprachenpaketen für Anwendungen scannt.	17. April 2023
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Erklärung hinzugefügt. Die neuen Anweisungen ermöglichen es Amazon Inspector, Scans des Entwicklercodes in AWS Lambda Funktionen anzufordern und Scandaten von Amazon CodeGuru Security zu empfangen. Darüber hinaus hat Amazon Inspector Berechtigungen zur Überprüfung von IAM-Richtlinien hinzugefügt. Amazon Inspector verwendet diese Informationen, um Lambda-Funktionen auf Code-Schwachstellen zu überprüfen.	28. Februar 2023
Neues Feature	Amazon Inspector bietet zusätzliche Unterstützung für Lambda-Funktionen in Form von Lambda-Code-Scans, die den Entwicklercode Ihrer Lambda-Funktionen auf Sicherheitslücken scannen.	28. Februar 2023

Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Erklärung hinzugefügt. Die neue Anweisung ermöglicht Amazon Inspector, Informationen CloudWatch darüber abzurufen, wann eine AWS Lambda Funktion zuletzt aufgerufen wurde. verwendet diese Informationen, um Scans auf die Lambda-Funktionen in Ihrer Umgebung zu konzentrieren, die in den letzten 90 Tagen aktiv waren.	20. Februar 2023
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Erklärung hinzugefügt. Die neue Erklärung ermöglicht es Amazon Inspector, Informationen über Ihre AWS Lambda Funktionen abzurufen. Amazon Inspector verwendet diese Informationen, um Ihre Lambda-Funktionen auf Sicherheitslücken zu überprüfen.	28. November 2022
Neues Feature	Amazon Inspector bietet Unterstützung für AWS Lambda Scanfunktionen.	28. November 2022

Aktualisierter Inhalt	Es wurden Verfahren, Richtlinienbeispiele und Tipps für den Export von Ergebnisberichten aus Amazon Inspector in einen Amazon Simple Storage Service (Amazon S3) -Bucket hinzugefügt.	14. Oktober 2022
Neuer Inhalt	Es wurden Informationen zur Bewertung der Amazon Inspector Inspector-Abdeckung Ihrer AWS Umgebung mithilfe der Amazon Inspector Inspector-Konsole hinzugefügt. Die Informationen umfassen Beschreibungen der Statuswerte für einzelne Ressourcen in Ihrer Umgebung.	7. Oktober 2022

Neues Feature	Amazon Inspector bietet jetzt zusätzliche Informationen zur Behebung von Sicherheitslücken in Paketen. Zu den Suchdetails wurden neue Felder hinzugefügt. Die neuen Felder geben Aufschluss darüber, ob ein Update im Rahmen eines Paket-Updates verfügbar ist. Wenn ein Update verfügbar ist, werden im Abschnitt „Vorgeschlagene Abhilfemaßnahmen“ eines Ergebnisses die Befehle angezeigt, die Sie ausführen können, um das Problem zu beheben.	02. September 2022
Aktualisierte Funktionalität	Amazon Inspector hat der AmazonInspector2ServiceRolePolicyRichtlinie eine neue Aktion hinzugefügt. Die neue Aktion ermöglicht es Amazon Inspector, SSM-Zuordnungsausführungen zu beschreiben. Amazon Inspector hat außerdem zusätzlichen Ressourcenbereich hinzugefügt, damit Amazon Inspector SSM-Verknüpfungen mit `AmazonInspector2` eigenen SSM-Dokumenten erstellen, aktualisieren, löschen und starten kann.	31. August 2022

Neues Feature	Amazon Inspector unterstützt jetzt Scans für Windows Instances. Amazon Inspector kann jetzt SSM-verwaltete Instances scannen, auf denen unterstützte Windows Betriebssysteme ausgeführt werden. Scans von Windows Hosts werden vom Amazon Inspector SSM-Plugin durchgeführt, das über neue SSM-Verknüpfungen installiert und aufgerufen wird, die automatisch von Amazon Inspector erstellt werden.	31. August 2022
Aktualisierte Funktionalität	Amazon Inspector hat den Ressourcenbereich der AmazonInspector2ServiceRolePolicyRichtlinie aktualisiert, sodass Amazon Inspector Softwareinventar in anderen AWS Partitionen erfassen kann.	12. August 2022
Aktualisierte Funktionalität	In der AmazonInspector2ServiceRolePolicyRichtlinie hat Amazon Inspector den Ressourcenbereich der Aktionen neu strukturiert, sodass Amazon Inspector SSM-Verknüpfungen erstellen, löschen und aktualisieren kann.	10. August 2022

Neues Feature	Amazon Inspector unterstützt jetzt die Änderung Ihrer Einstellung für die Dauer Ihres automatisierten ECR-Rescans. Die Einstellung für die Dauer des automatischen erneuten Scans in Amazon ECR bestimmt, wie lange Amazon Inspector kontinuierlich Bilder überwacht, die in Repositorys übertragen werden. Wenn ein Bild älter als die Scandauer ist, scannt Amazon Inspector das Bild nicht mehr und schließt alle vorhandenen Ergebnisse dafür. Bei allen neuen Konten wird die Dauer des automatischen ECR-Wiederholungsscans automatisch auf „Lebenszeit“ gesetzt. Zuvor erstellte Konten hatten eine Dauer von 30 Tagen für den automatischen ECR-Rescan, aber Sie können jetzt zwischen einer Dauer von 30 Tagen, 180 Tagen oder lebenslangen Scans wählen.	25. Juni 2022

Neue Funktionalität	Amazon Inspector hat eine neue AWS verwaltete Richtlinie hinzugefügt, die den AmazonInspector2ReadOnlyAccess schreibgeschützten Zugriff auf die Funktionen von Amazon Inspector ermöglicht.	21. Januar 2022
Allgemeine Verfügbarkeit	Dies ist die erste öffentliche Version des Amazon Inspector Inspector-Benutzerhandbuchs.	29. November 2021

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.