aws

사용 설명서

Amazon Inspector –

Amazon Inspector –: 사용 설명서

Table of Contents

Amazon Inspector란 무엇인가? 1
특성 1
Amazon Inspector 액세스 3
시작하기 5
Amazon Inspector를 활성화하기 전에 5
시작하기 자습서: Amazon Inspector 활성화 6
자동 스캔 11
Amazon Inspector 스캔 유형 개요 11
스캔 유형 활성화 12
스캔 활성화 13
Amazon EC2 인스턴스 스캔 14
에이전트 기반 스캔 15
에이전트 없는 스캔 19
스캔 모드 관리 21
Amazon Inspector 스캔에서 인스턴스 제외 22
지원되는 운영 체제 22
Linux 인스턴스에 대한 심층 검사 23
Windows EC2 인스턴스 스캔 27
Amazon ECR 컨테이너 이미지 스캔 30
Amazon ECR 스캔의 스캔 동작 31
실행 중인 컨테이너에 컨테이너 이미지 매핑 32
지원되는 운영 체제 및 미디어 유형 33
Amazon ECR 재스캔 기간 구성 34
Lambda 함수 스캔 36
Lambda 함수 스캔의 스캔 동작 37
지원되는 런타임 및 함수 37
Amazon Inspector Lambda 표준 스캔 38
Amazon Inspector Lambda 코드 스캔 39
스캔 유형 비활성화 41
스캔 비활성화 42
CIS 스캔 44
Amazon Inspector CIS 스캔을 위한 Amazon EC2 인스턴스 요구 사항 45
프라이빗 Amazon EC2 인스턴스에서 CIS 스캔을 실행하기 위한 Amazon Virtual Private Cloud 엔드포인트 요구 사항 46

CIS 스캔 실행 46
를 사용하여 Amazon Inspector CIS 스캔을 관리하기 위한 고려 사항 AWS Organizations 47
Amazon Inspector CIS 스캔에 사용되는 Amazon Inspector 소유 Amazon S3 버킷 48
CIS 스캔 구성 생성 50
CIS 스캔 결과 보기 51
CIS 스캔 구성 편집 52
CIS 스캔 결과 다운로드 52
Amazon Inspector 코드 보안 54
사전 조건 54
코드 보안 활성화 54
액세스할 고객 관리형 키 생성 AWS KMS 54
통합 생성 57
GitHub 통합 생성 58
GitLab Self Managed 통합 생성 59
통합 보기 61
코드 리포지토리 보기 62
통합 삭제 63
스캔 구성 생성 63
스캔 구성 보기 65
스캔 구성 편집 66
스캔 구성 삭제 67
온디맨드 스캔 수행 67
지원되는 언어 67
코드 보안 비활성화 69
조사 결과 이해 70
조사 결과 유형 71
패키지 취약성 71
코드 취약성 71
네트워크 연결성 72
조사 결과 보기 73
조사 결과 세부 정보 보기 74
Amazon Inspector 점수 보기 77
Amazon Inspector 점수 78
취약성 인텔리전스 79
조사 결과의 심각도 수준 이해 80
소프트웨어 패키지 취약성 심각도 80

코드 취약성 심각도 81
네트워크 연결성 심각도 80
조사 결과 관리 84
조사 결과 필터링 84
Amazon Inspector 콘솔에서 필터 생성 84
조사 결과 안 보이게 하기 85
억제 규칙 생성 85
숨겨진 조사 결과 보기 86
억제 규칙 편집 86
억제 규칙 삭제 87
조사 결과 보고서 내보내기 87
1단계: 권한 확인 88
2단계: S3 버킷 구성 90
3단계: 구성 AWS KMS key 93
4단계: 조사 결과 보고서 구성 및 내보내기 96
오류 해결 99
EventBridge를 사용하여 조사 결과에 대한 응답 자동화 99
이벤트 스키마 100
Amazon Inspector 조사 결과를 알리는 EventBridge 규칙 생성 102
Amazon Inspector 다중 계정 환경용 EventBridge 106
대시보드 107
대시보드 보기 107
대시보드 구성 요소 이해 108
취약성 데이터베이스 검색 111
취약성 데이터베이스 검색 111
CVE 세부 정보 이해 112
CVE 세부 정보 112
취약성 인텔리전스 112
참조 112
SBOM 내보내기 113
Amazon Inspector 형식 113
SBOM용 필터 118
SBOM 구성 및 내보내기 119
EventBridge 스키마 122
Amazon Inspector의 Amazon EventBridge 기본 스키마 122
Amazon Inspector 조사 결과 이벤트 스키마 예제 123

Amazon Inspector 최초 스캔 완료 이벤트 스키마 예제 135
Amazon Inspector 적용 범위 이벤트 스키마 예제 138
Amazon Inspector 스키마 자동 활성화 예제 139
SSM 플러그인 140
Linux용 Amazon Inspector SSM 플러그인 140
Amazon Inspector SSM 플러그인 제거 140
Windows용 Amazon Inspector SSM 플러그인 140
Amazon Inspector SSM 플러그인 제거 141
Amazon Inspector SBOM 생성기 142
지원되는 패키지 유형 142
지원되는 컨테이너 이미지 구성 검사 142
Sbomgen 설치하기 143
Sbomgen 사용하기 144
컨테이너 이미지에 대한 SBOM 생성 및 결과 출력 144
디렉터리 및 아카이브에서 SBOM 생성 145
Go 또는 Rust 컴파일된 바이너리에서 SBOM 생성 146
탑재된 볼륨에서 SBOM 생성 146
취약성 식별을 위해 Amazon Inspector로 SBOM 전송 147
추가 스캐너를 사용하여 탐지 기능 향상 149
스캔할 최대 파일 크기를 조정하여 컨테이너 스캔 최적화 150
진행률 표시기 비활성화 150
Sbomgen을 사용하여 프라이빗 레지스트리에 인증 151
캐시된 자격 증명을 사용하여 인증(권장) 151
대화형 방법을 사용한 인증 151
비대화형 방법을 사용한 인증 152
Sbomgen의 예시 출력 152
이전 버전 155
운영 체제 컬렉션 166
지원되는 운영 체제 아티팩트 166
APK 기반 OS 패키지 컬렉션 167
DPKG 기반 OS 패키지 컬렉션 168
RPM 기반 OS 패키지 컬렉션 170
Windows OS 버전 컬렉션 171
Chainguard 이미지 패키지 컬렉션 172
Distroless 이미지 패키지 컬렉션 173
MinimOS 패키지 컬렉션 174

종속성 컬렉션 175
Go 종속성 스캔 175
Java 종속성 스캔 178
JavaScript 종속성 스캔 182
.NET 종속성 스캔 188
PHP 종속성 스캔 194
Python 종속성 스캔 196
Ruby 종속성 스캔 201
Rust 종속성 스캔 204
지원되지 않는 아티팩트 207
에코시스템 컬렉션 208
지원되는 에코시스템 208
7-Zip 에코시스템 컬렉션 211
Apache 에코시스템 컬렉션 212
Atlassian 에코시스템 컬렉션 215
Curl 에코시스템 컬렉션 217
Elasticsearch 에코시스템 컬렉션 219
Google 에코시스템 컬렉션 220
Java 에코시스템 컬렉션 222
Jenkins 에코시스템 컬렉션 224
MariaDB 및 MySQL 에코시스템 컬렉션 225
Microsoft applications 에코시스템 컬렉션 227
Nginx 에코시스템 컬렉션 231
Node.JS 런타임 컬렉션 233
OpenSSH 에코시스템 컬렉션 234
OpenSSL 에코시스템 컬렉션 235
Oracle Database Server 컬렉션 236
PHP 에코시스템 컬렉션 237
WordPress 에코시스템 컬렉션 238
SSL/TLS 인증서 스캔 241
Sbomgen 인증서 스캔 사용 241
라이선스 컬렉션 244
라이선스 정보 수집 245
지원되는 패키지 245
패키지 URL 252
PURL 구조 252

버전 참조 254
권장 사항 255
Java 255
JavaScript 255
Python 256
CycloneDX 네임스페이스 사용 256
amazon:inspector:sbom_scanner 네임스페이스 분류법 256
amazon:inspector:sbom_generator 네임스페이스 분류법 258
CI/CD 통합 263
플러그인 통합 263
지원되는 CI/CD 솔루션 264
사용자 지정 통합 264
CI/CD 통합을 위한 계정 설정 265
에 가입 AWS 계정 265
관리자 액세스 권한이 있는 사용자 생성 266
CI/CD 통합을 위한 IAM 역할 구성 267
Amazon Inspector Dockerfile 검사 268
Sbomgen Dockerfile 검사 사용 269
지원되는 Dockerfile 검사 271
사용자 지정 CI/CD 통합 생성 276
1단계. 구성 AWS 계정 276
2단계. Sbomgen 바이너리 설치 276
3단계. Sbomgen 사용하기 277
4단계. Amazon Inspector 스캔 API 직접 호출 277
(선택 사항) 5단계. 단일 명령으로 SBOM 생성 및 스캔 277
API 출력 형식 278
Jenkins 플러그인 285
1단계. 설정 AWS 계정 286
2단계. Amazon Inspector Jenkins 플러그인을 설치합니다. 286
(선택 사항) 3단계. Jenkins에 docker 자격 증명 추가 286
(선택 사항) 4단계. AWS 자격 증명 추가 287
5단계. Jenkins 스크립트에 CSS 지원 추가 287
6단계. 빌드에 Amazon Inspector 스캔 추가 287
7단계. Amazon Inspector 취약성 보고서 확인 292
문제 해결 293
TeamCity 플러그인 295

GitHub 작업 ... 297
GitLab 구성 요소 ... 297
CodeCatalyst 작업 사용 ... 298
Amazon Inspector 스캔 작업 사용 ... 298
적용 범위 평가 ... 299
계정 수준 적용 범위 평가 ... 300
Amazon EC2 인스턴스의 적용 범위 평가 ... 300
Amazon EC2 인스턴스 상태 값 ... 301
Amazon ECR 리포지토리의 적용 범위 평가 ... 303
Amazon ECR 리포지토리 스캔 상태 값 ... 304
Amazon ECR 컨테이너 이미지의 적용 범위 평가 ... 305
Amazon ECR 컨테이너 이미지 스캔 상태 값 ... 305
AWS Lambda 함수 적용 범위 평가 ... 307
Lambda 함수 스캔 상태 값 ... 307
다중 계정 관리 ... 309
위임된 관리자 계정 및 멤버 계정 이해 ... 309
조직 정책 거버넌스 모델 ... 309
위임 관리자 작업 ... 310
멤버 계정 작업 ... 311
관리자 계정 지정하기 ... 312
고려 사항 ... 312
위임 관리자를 지정하는 데 필요한 권한 ... 313
위임 관리자 지정 ... 314
멤버 계정에 대한 Amazon Inspector 스캔 활성화 ... 315
멤버 계정 연결 해제 ... 319
위임된 관리자 제거 ... 320
리소스에 태그 지정 ... 322
태그 지정 기본 사항 ... 322
태그 추가 ... 323
Amazon Inspector 리소스에 태그 추가 ... 323
태그 제거 ... 324
Amazon Inspector 리소스에서 태그 제거 ... 324
사용법 ... 326
사용량 콘솔 사용 ... 326
Amazon Inspector의 사용 비용 계산 방식 이해 ... 327
Amazon Inspector 무료 평가판 정보 ... 328

보안 329
데이터 보호 330
저장된 데이터 암호화 331
전송 중 암호화 335
자격 증명 및 액세스 관리 335
대상 336
ID를 통한 인증 336
정책을 사용하여 액세스 관리 338
Amazon Inspector에서 IAM을 사용하는 방법 339
ID 기반 정책 예시 344
AWS 관리형 정책 348
서비스 연결 역할 사용 362
문제 해결 371
Amazon Inspector 모니터링 372
CloudTrail 로그 373
규정 준수 확인 376
복원력 376
인프라 보안 377
인시던트 대응 377
AWS PrivateLink 377
고려 사항 378
인터페이스 엔드포인트 생성 378
통합 379
에서 Amazon Inspector 사용 AWS Organizations 379
Amazon Inspector와 Amazon ECR 통합 379
Security Hub CSPM과 Amazon Inspector 통합 380
Amazon ECR 통합 380
통합 활성화 380
다중 계정 환경과의 통합 사용 380
Security Hub CSPM 통합 381
에서 Amazon Inspector 조사 결과 보기 AWS Security Hub CSPM 381
Security Hub CSPM과의 Amazon Inspector 통합 활성화 및 구성 385
조직 정책을 사용하여 Security Hub CSPM에서 Amazon Inspector 활성화 385
통합에서 조사 결과의 흐름 비활성화 386
Security Hub CSPM에서 Amazon Inspector에 대한 보안 제어 보기 386
지원되는 운영 체제 및 프로그래밍 언어 387

지원되는 운영 체제 388
지원되는 운영 체제: Amazon EC2 스캔 388
지원되는 운영 체제: Amazon Inspector를 사용한 Amazon ECR 스캔 391
지원되는 운영 체제: CIS 스캔 394
지원되는 운영 체제: Amazon Inspector Scan API 395
중단된 운영 체제 397
지원되는 프로그래밍 언어 401
지원되는 프로그래밍 언어: Amazon EC2 에이전트 없는 스캔 401
지원되는 프로그래밍 언어: Amazon EC2 심층 검사 402
지원되는 프로그래밍 언어: Amazon ECR 스캔 402
지원되는 런타임 403
지원되는 런타임: Amazon Inspector Lambda 표준 스캔 403
지원되는 런타임: Amazon Inspector Lambda 코드 스캔 405
Amazon Inspector 비활성화 406
조직 정책에서 관리하는 Amazon Inspector 비활성화 407
Amazon Inspector 비활성화 408
할당량 409
리전 및 엔드포인트 410
Amazon Inspector의 서비스 엔드포인트 410
Amazon Inspector 스캔 API용 엔드포인트 410
리전별 특성 가용성 417
문서 기록 422
Amazon Inspector 제품 업데이트 422
Amazon Inspector Security Research 447
감지 요약 447
최근 악성 패키지 보고서(지난 10개) 447
AWS 용어집 449
..... cdl

Amazon Inspector란 무엇인가?

Amazon Inspector는 워크로드를 자동으로 검색하고 소프트웨어 취약성 및 의도하지 않은 네트워크 노출이 있는지 지속적으로 스캔하는 취약성 관리 서비스입니다. Amazon Inspector는 Amazon EC2 인스턴스, Amazon ECR의 컨테이너 이미지 및 Lambda 함수를 검색하고 스캔합니다. Amazon Inspector에서 소프트웨어 취약성 또는 의도하지 않은 네트워크 노출을 탐지하면 문제에 대한 세부 보고서인 조사 결과가 생성됩니다. Amazon Inspector 콘솔 또는 API에서 조사 결과를 관리할 수 있습니다.

> **Note**
>
> 지원 요청을 제출할 때 Amazon Inspector는 문제가 해결되도록 저장된 (동일 AWS 리전 한 지역 내에 있는)의 관련 결과에 액세스하고 처리할 수 있습니다.

주제

- Amazon Inspector의 특성
- Amazon Inspector 액세스

Amazon Inspector의 특성

여러 Amazon Inspector 계정을 중앙에서 관리

AWS 환경에 여러 계정이 있는 경우 AWS Organizations를 사용하여 단일 계정을 통해 환경을 중앙에서 관리할 수 있습니다. 이 방법을 사용하면 특정 계정을 Amazon Inspector의 위임 관리자 계정으로 지정할 수 있습니다.

한 번의 클릭으로 전체 조직에 대해 Amazon Inspector를 활성화할 수 있습니다. 또한 향후 멤버가 조직에 가입할 때마다 서비스를 자동으로 활성화할 수 있습니다. Amazon Inspector 위임 관리자 계정은 조직 멤버에 대한 조사 결과 데이터와 특정 설정을 관리할 수 있습니다. 여기에는 모든 멤버 계정에 대한 집계된 조사 결과 세부 정보 보기, 멤버 계정에 대한 스캔 활성화 또는 비활성화, AWS 조직 내에서 스캔한 리소스 검토가 포함됩니다.

환경의 취약성과 네트워크 노출 여부를 지속적으로 스캔

Amazon Inspector를 사용하면 평가 스캔을 수동으로 예약하거나 구성할 필요가 없습니다. Amazon Inspector는 적합한 리소스를 자동으로 검색하고 스캔을 시작합니다. Amazon Inspector는 EC2 인스턴

스에 새 패키지를 설치하거나, 패치를 설치하거나, 리소스에 영향을 미치는 새로운 일반적인 취약성 및 노출(CVE)이 발표되는 경우 등 새로운 취약성을 유발할 수 있는 변경 사항에 대응하여 리소스를 자동으로 재스캔함으로써 리소스 수명 주기 전반에 걸쳐 환경을 계속 평가합니다. 기존 보안 스캔 소프트웨어와 달리 Amazon Inspector는 플릿 성능에 미치는 영향을 최소화합니다.

취약성 또는 오픈 네트워크 경로가 식별되면 Amazon Inspector에서는 사용자가 조사할 수 있도록 조사 결과를 생성합니다. 조사 결과에는 취약성, 영향을 받는 리소스 및 해결 권장 사항에 대한 포괄적인 세부 정보가 포함됩니다. 조사 결과를 적절하게 수정하면 Amazon Inspector에서 자동으로 수정 사항을 탐지하고 조사 결과를 종결합니다.

Amazon Inspector 위험 점수를 사용하여 정확하게 취약성 평가

Amazon Inspector는 스캔을 통해 환경에 대한 정보를 수집하므로 사용 환경에 맞게 특별히 조정된 심각도 점수를 제공합니다. Amazon Inspector는 취약성에 대한 NVD(National Vulnerability Database) 기본 점수를 구성하는 보안 지표를 검사하여 컴퓨팅 환경에 따라 조정합니다. 예를 들어 네트워크를 통해 취약성이 악용될 소지가 있지만 인터넷으로 연결되는 오픈 네트워크 경로를 인스턴스에서 사용할 수 없는 경우 이 서비스가 Amazon EC2 인스턴스의 Amazon Inspector 조사 결과 점수를 낮출 수 있습니다. 이 점수는 CVSS 형식이며 NVD에서 제공하는 기본 CVSS(Common Vulnerability Scoring System) 점수를 수정한 것입니다.

Amazon Inspector 대시보드를 사용하여 영향력이 큰 조사 결과 식별

Amazon Inspector 대시보드에서는 환경 전반의 조사 결과를 개괄적으로 볼 수 있습니다. 대시보드에서는 조사 결과의 세부 정보에 액세스할 수 있습니다. 대시보드에는 사용 환경의 스캔 적용 범위, 가장 중요한 조사 결과, 가장 많은 조사 결과가 발견된 리소스에 대한 간소화된 정보가 포함되어 있습니다. Amazon Inspector 대시보드의 위험에 기반한 해결 방법 패널에는 가장 많은 수의 인스턴스와 이미지에 영향을 미치는 조사 결과가 표시됩니다. 이 패널에서는 환경에 가장 큰 영향을 미치는 조사 결과를 쉽게 식별하고, 조사 결과 세부 정보를 검토하고, 제안된 솔루션을 검토할 수 있습니다.

사용자 지정 가능한 보기를 사용하여 조사 결과 관리

대시보드 외에도 Amazon Inspector 콘솔에서 조사 결과 보기를 제공합니다. 이 페이지에는 사용 환경에 대한 모든 조사 결과가 나열되고 개별 조사 결과에 대한 세부 정보가 제공됩니다. 범주 또는 취약성 유형별로 그룹화된 조사 결과를 볼 수 있습니다. 각 보기에서는 필터를 사용하여 결과를 추가로 사용자 지정할 수 있습니다. 필터를 사용하여 원치 않는 조사 결과를 보기에서 숨기는 억제 규칙을 생성할 수도 있습니다.

필터와 억제 규칙을 사용하여 모든 조사 결과 또는 사용자 지정된 조사 결과를 보여주는 조사 결과 보고서를 생성할 수 있습니다. 보고서는 CSV 또는 JSON 형식으로 생성할 수 있습니다.

다른 서비스 및 시스템과 함께 조사 결과 모니터링 및 처리

다른 서비스 및 시스템과의 통합을 지원하기 위해 Amazon Inspector는 조사 결과를 Amazon EventBridge에 조사 결과 이벤트로 게시합니다. EventBridge는 결과 데이터를 AWS Lambda 함수 및 Amazon Simple Notification Service(Amazon SNS) 주제와 같은 대상으로 라우팅할 수 있는 서버리스 이벤트 버스 서비스입니다. EventBridge를 사용하면 기존 보안 및 규정 준수 워크플로우의 일부로 조사 결과를 거의 실시간으로 모니터링하고 처리할 수 있습니다.

를 활성화한 경우 AWS Security Hub CSPM Amazon Inspector는 Security Hub CSPM에도 조사 결과를 게시합니다. Security Hub CSPM은 AWS 환경 전반의 보안 태세에 대한 포괄적인 보기를 제공하고 보안 업계 표준 및 모범 사례를 기준으로 환경을 확인하는 데 도움이 되는 서비스입니다. Security Hub CSPM을 사용하면 조직의 보안 태세에 대한 광범위한 분석의 일환으로 조사 결과를 보다 쉽게 모니터링하고 처리할 수 있습니다 AWS.

Amazon Inspector 액세스

Amazon Inspector는 대부분에서 사용할 수 있습니다 AWS 리전. 현재 Amazon Inspector가 사용 가능한 모든 리전 목록은 Amazon Web Services 일반 참조에서 Amazon Inspector 엔드포인트 및 할당량을 참조하세요. AWS 리전에 대해 자세히 알아보려면 Amazon Web Services 일반 참조에서 AWS 리전관리를 참조하세요. 각 리전에서 다음과 같은 방법으로 Amazon Inspector를 사용할 수 있습니다.

AWS 관리 콘솔

는 리소스를 생성하고 관리하는 AWS 데 사용할 수 있는 브라우저 기반 인터페이스 AWS Management Console 입니다. Amazon Inspector 콘솔은 해당 콘솔의 일부로 Amazon Inspector 계정 및 리소스에 대한 액세스를 제공합니다. Amazon Inspector 콘솔에서 Amazon Inspector 작업을 수행할 수 있습니다.

AWS 명령줄 도구

AWS 명령줄 도구를 사용하면 시스템의 명령줄에서 명령을 실행하여 Amazon Inspector 작업을 수행할 수 있습니다. 명령줄을 사용하는 것이 콘솔을 사용하는 것보다 더 빠르고 편리할 수 있습니다. 작업을 수행하는 스크립트를 작성할 때도 명령줄 도구가 유용합니다.

AWS 는 AWS Command Line Interface (AWS CLI)와의 두 가지 명령줄 도구 세트를 제공합니다 AWS Tools for PowerShell. 설치 및 사용에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서를 AWS CLI참조하세요. Tools for PowerShell 설치 및 사용에 대한 자세한 내용은 AWS Tools for PowerShell 사용 설명서를 참조하세요.

AWS SDK

AWS 는 Java, Go, Python, C++ 및 .NET을 비롯한 다양한 프로그래밍 언어 및 플랫폼을 위한 라이브러리 및 샘플 코드로 구성된 SDKs를 제공합니다. SDK를 사용하면 Amazon Inspector 및 다른 AWS 서비스에 프로그래밍 방식으로 편리하게 액세스할 수 있습니다. SDK는 요청에 암호화 방식으로 서명, 오류 관리 및 자동으로 요청 재시도와 같은 작업을 포함합니다. AWS SDKs. AWS

Amazon Inspector REST API

Amazon Inspector REST API는 Amazon Inspector 계정 및 리소스에 대한 포괄적인 프로그래밍 방식의 액세스를 제공합니다. 이 API를 사용하면 HTTPS 요청을 Amazon Inspector로 직접 보낼 수 있습니다. 그러나 AWS 명령줄 도구 및 SDKs와 달리이 API를 사용하려면 애플리케이션이 요청에 서명하기 위한 해시 생성과 같은 하위 수준 세부 정보를 처리해야 합니다.

Amazon Inspector 시작하기

이 단원에서는 Amazon Inspector를 활성화하기 전에 고려해야 할 정보와 Amazon Inspector 콘솔 및 Amazon Inspector API를 사용하여 Amazon Inspector를 활성화하고 조사 결과를 확인하는 방법을 설명하는 시작하기 자습서를 제공합니다.

주제

- Amazon Inspector를 활성화하기 전에
- 시작하기 자습서: Amazon Inspector 활성화

Amazon Inspector를 활성화하기 전에

Amazon Inspector를 활성화하기 전에 다음 사항을 고려하세요.

Amazon Inspector는 리전 서비스임

데이터는 Amazon Inspector를 활성화 AWS 리전 하는에 저장됩니다. Amazon Inspector를 사용하려는 모든에 대해 시작하기 자습서의 첫 번째 부분에 AWS 리전 있는 단계를 반복합니다.

Amazon Inspector에서 서비스 연결 역할 AWSServiceRoleForAmazonInspector2 및 AWSServiceRoleForAmazonInspector2Agentless를 생성함

서비스 연결 역할은 AWS 서비스에 연결된 AWS Identity and Access Management (IAM)의 역할입니다. AWSServiceRoleForAmazonInspector2 및 AWSServiceRoleForAmazonInspector2Agentless를 사용하면 Amazon Inspector가 보안 평가를 수행하는 데 AWS 서비스 필요한에 액세스할 수 있습니다.

관리자 권한이 있는 IAM 자격 증명으로 Amazon Inspector를 활성화할 수 있음

IAM 또는 AWS IAM Identity Center를 통해 사용자를 생성하여 자격 증명을 보호합니다. 이렇게 하면 Amazon Inspector를 관리하는 데 필요한 권한만 사용자에게 부여할 수 있습니다. 자세한 내용은 AWS 관리형 정책: Amazon AmazonInspectorFullAccess를 참조하세요.

하이브리드 스캔이 자동으로 활성화됨

하이브리드 스캔에는 에이전트 기반 스캔 및 에이전트 없는 스캔이 포함됩니다. 기본적으로 Amazon Inspector는 모든 적격 Amazon EC2 인스턴스에 대해 이러한 스캔 방법을 사용합니다. 자세한 내용은 Amazon Inspector로 Amazon EC2 인스턴스 스캔을 참조하세요.

Amazon ECR 스캔 및 Lambda 함수 스캔에는 SSM 에이전트가 필요하지 않습니다.

에이전트 기반 스캔의 경우 SSM 에이전트를 사용하여 소프트웨어 인벤토리를 수집합니다. 에이전트 없는 스캔의 경우 Amazon EBS 스냅샷을 사용하여 소프트웨어 인벤토리를 수집합니다.

> **Note**
>
> 기본적으로 SSM 에이전트는 Amazon Machine Images를 기반으로 Amazon EC2 인스턴스에 이미 설치되어 있습니다. 그러나 경우에 따라 SSM 에이전트를 수동으로 활성화해야 할 수도 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 SSM Agent 작업을 참조하세요.

월별 비용은 스캔한 워크로드를 기준으로 합니다.

자세한 내용은 Amazon Inspector 요금을 참조하세요.

를 사용한 다중 계정 활성화 AWS Organizations

를 사용하는 조직의 경우 AWS Organizations Amazon Inspector는 위임된 관리자 관리와 조직 정책 기반 활성화를 모두 지원합니다. 조직 정책은 새 계정에 대한 자동 활성화를 통해 중앙 집중식 거버넌스를 제공합니다. 두 접근 방식에 대한 자세한 지침은 섹션을 참조하세요시작하기 자습서: Amazon Inspector 활성화.

시작하기 자습서: Amazon Inspector 활성화

이 주제에서는 독립 실행형 계정 환경(멤버 계정) 및 다중 계정 환경(위임된 관리자 계정)에 대해 Amazon Inspector를 활성화하는 방법을 설명합니다. Amazon Inspector를 활성화하면 워크로드를 자동으로 검색하고 소프트웨어 취약성 및 의도하지 않은 네트워크 노출이 있는지 스캔하기 시작합니다.

Standalone account environment

다음 절차에서는 콘솔에서 멤버 계정에 대해 Amazon Inspector를 활성화하는 방법을 설명합니다. 프로그래밍 방식으로 Amazon Inspector를 활성화하려면 inspector2-enablement-with-cli를 사용합니다.

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 시작을 선택합니다.
3. Amazon Inspector 활성화를 선택합니다.

독립 실행형 계정에 대해 Amazon Inspector를 활성화하면 모든 스캔 유형이 기본적으로 활성화됩니다. 멤버 계정에 대한 자세한 내용은 Amazon Inspector의 위임된 관리자 계정 및 멤버 계정 이해를 참조하세요.

Multi-account (with AWS Organizations policy)

AWS Organizations 정책은 조직 전체에서 Amazon Inspector를 활성화하기 위한 중앙 집중식 거버넌스를 제공합니다. 조직 정책을 사용하면 정책이 적용되는 모든 계정에 대해 Amazon Inspector 활성화가 자동으로 관리되며, 멤버 계정은 Amazon Inspector API를 사용하여 정책 관리형 스캔을 수정할 수 없습니다.

사전 조건

- 계정은 AWS Organizations 조직의 일부여야 합니다.
- 에서 조직 정책을 생성하고 관리할 수 있는 권한이 있어야 합니다 AWS Organizations.
- Amazon Inspector에 대한 신뢰할 수 있는 액세스를 활성화해야 합니다 AWS Organizations. 지침은 AWS Organizations 사용 설명서의 Amazon Inspector에 대한 신뢰할 수 있는 액세스 활성화를 참조하세요.
- Amazon Inspector 서비스 연결 역할은 관리 계정에 있어야 합니다. 생성하려면 관리 계정에서 Amazon Inspector를 활성화하거나 관리 계정에서 다음 명령을 실행합니다.
  - `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`
- Amazon Inspector 위임된 관리자를 지정해야 합니다.

> **Note**
>
> 관리 계정 및 위임된 관리자의 서비스 연결 Amazon Inspector 역할이 없으면 조직 정책은 Amazon Inspector 활성화를 적용하지만 멤버 계정은 중앙 집중식 조사 결과 및 계정 관리를 위해 Amazon Inspector 조직과 연결되지 않습니다.

AWS Organizations 정책을 사용하여 Amazon Inspector를 활성화하려면

1. 중앙 집중식 조사 결과 가시성을 위해 멤버 계정이 Amazon Inspector 조직과 연결되도록 조직 정책을 생성하기 전에 Amazon Inspector의 위임된 관리자를 지정합니다. AWS Organizations

관리 계정에 로그인하고 https://console.aws.amazon.com/inspector/v2/home Amazon Inspector 콘솔을 연 다음의 단계를 따릅니다AWS 조직의 위임된 관리자 지정.

> **Note**
>
> AWS Organizations Amazon Inspector 위임된 관리자 계정 ID와 Amazon Inspector 지정 위임된 관리자 계정 ID를 동일하게 유지하는 것이 좋습니다. AWS Organizations 위임된 관리자 계정 ID가 Amazon Inspector 위임된 관리자 계정 ID와 다른 경우 Amazon Inspector는 Inspector 지정 계정 ID의 우선 순위를 지정합니다. Amazon Inspector 위임된 관리자가 설정되지 않았지만 AWS Organizations 위임된 관리자가 설정되고 관리 계정에 Amazon Inspector 서비스 연결 역할이 있는 경우 Amazon Inspector는 AWS Organizations 위임된 관리자 계정 ID를 Amazon Inspector 위임된 관리자로 자동으로 할당합니다.

2. Amazon Inspector 콘솔에서 관리 계정의 일반 설정으로 이동합니다. 위임 정책에서 문 연결을 선택합니다. 정책 설명 연결 대화 상자에서 정책을 검토하고 정책을 검토했으며 정책이 부여하는 권한을 이해했음을 확인함을 선택한 다음 설명 연결을 선택합니다.

> **Important**
>
> 위임 정책 설명을 연결하려면 관리 계정에 다음 권한이 있어야 합니다.
>
> - AmazonInspector2FullAccess_v2 관리형 정책의 Amazon Inspector 권한 AmazonInspector2FullAccess
> - AWS Organizations `organizations:PutResourcePolicy` AWSOrganizationsFullAccess 관리형 정책의 권한
>
> `organizations:PutResourcePolicy` 권한이 누락된 경우 작업이 실패하고 오류가 발생합니다`Failed to attach statement to the delegation policy.`

3. 다음으로 Amazon Inspector AWS Organizations 정책을 생성합니다. 탐색 창에서 관리를 선택한 다음 구성을 선택합니다.
4. 취약성 관리 정책을 구성합니다. 정책에 대한 이름 및 설명(선택 사항)과 함께 세부 정보를 제공합니다.
5. Inspector 구성 페이지의 세부 정보 섹션에 정책의 이름과 설명을 입력합니다. 기능 선택에서 다음 중 하나를 수행합니다.

- 모든 기능 구성 및 활성화(권장)를 선택합니다. 그러면 EC2, ECR, Lambda 표준, Lambda 코드 스캔 및 코드 보안을 포함한 모든 Inspector 기능이 활성화됩니다.
- 기능의 하위 집합 선택을 선택합니다. 활성화해야 하는 스캔 유형 기능을 선택합니다.

6. 계정 선택 섹션에서 다음 옵션 중 하나를 선택합니다.

   - 구성을 모든 조직 단위 및 계정에 적용하려면 모든 조직 단위 및 계정을 선택합니다.
   - 구성을 특정 조직 단위 및 계정에 적용하려면 특정 조직 단위 및 계정을 선택합니다. 이 옵션을 선택하는 경우 검색 창 또는 조직 구조 트리를 사용하여 정책을 적용할 조직 단위 및 계정을 지정합니다.
   - 구성을 조직 단위 또는 계정에 적용하지 않으려면 조직 단위 또는 계정 없음을 선택합니다.

7. 리전 섹션에서 모든 리전 활성화, 모든 리전 비활성화 또는 리전 지정을 선택합니다.

   - 모든 리전 활성화를 선택하는 경우 새 리전을 자동으로 활성화할지 여부를 결정할 수 있습니다.
   - 모든 리전 비활성화를 선택하는 경우 새 리전을 자동으로 비활성화할지 여부를 결정할 수 있습니다.
   - 리전 지정을 선택하는 경우 활성화 및 비활성화할 리전을 선택해야 합니다.

   (선택 사항) 고급 설정은의 지침을 참조하세요 AWS Organizations.

   (선택 사항) 리소스 태그의 경우 구성을 쉽게 식별할 수 있도록 태그를 키-값 페어로 추가합니다.

8. 다음을 선택하고 변경 사항을 검토한 다음 적용을 선택합니다. 대상 계정이 정책에 따라 구성됩니다. 정책의 구성 상태가 정책 페이지 상단에 표시됩니다. 각 기능은 구성되었는지 또는 배포 실패가 있는지에 대한 상태를 제공합니다. 실패의 경우 실패 메시지의 링크를 선택하여 자세한 내용을 확인합니다. 계정 수준에서 유효한 정책을 보려면 구성 페이지에서 계정을 선택할 수 있는 조직 탭을 검토할 수 있습니다.

조직 정책을 통해 Amazon Inspector를 활성화하면 정책이 적용되는 계정은 Amazon Inspector API 또는 콘솔을 통해 정책 관리형 스캔 유형을 비활성화할 수 없습니다. 위임된 관리자 및 멤버 계정이 조직 정책에 따라 수행할 수 있는 작업과 수행할 수 없는 작업에 대한 자세한 내용은 섹션을 참조하세요를 사용하여 Amazon Inspector에서 여러 계정 관리 AWS Organizations.

Multi-account (without AWS Organizations policy)

> **Note**
>
> 이 절차를 완료하려면 AWS Organizations 관리 계정을 사용해야 합니다. AWS Organizations 관리 계정만 위임된 관리자를 지정할 수 있습니다. 위임된 관리자를 지정하려면 권한이 필요할 수 있습니다. 자세한 내용은 위임 관리자를 지정하는 데 필요한 권한 단원을 참조하십시오.

Amazon Inspector를 처음 활성화하면 Amazon Inspector가 해당 계정에 대한 서비스 연결 역할 `AWSServiceRoleForAmazonInspector`를 생성합니다. Amazon Inspector에서 서비스 연결 역할을 사용하는 방법에 대한 자세한 내용은 Amazon Inspector에 서비스 연결 역할 사용 섹션을 참조하세요.

Amazon Inspector 위임 관리자를 지정하려면

1. AWS Organizations 관리 계정에 로그인한 다음 https://console.aws.amazon.com/inspector/v2/home Amazon Inspector 콘솔을 엽니다.
2. Get started를 선택합니다.
3. 위임된 관리자에서 위임된 관리자로 AWS 계정 지정할의 12자리 ID를 입력합니다.
4. 위임을 선택한 다음 위임을 다시 선택합니다.
5. (선택 사항) AWS Organizations 관리 계정에 대해 Amazon Inspector를 활성화하려면 서비스 권한에서 Amazon Inspector 활성화를 선택합니다.

위임된 관리자를 지정하면 기본적으로 계정에 대해 모든 스캔 유형이 활성화됩니다. 위임된 관리자 계정에 대한 자세한 내용은 Amazon Inspector의 위임된 관리자 계정 및 멤버 계정 이해를 참조하세요.

Amazon Inspector의 자동 스캔 유형

Amazon Inspector는 리소스에 실행 가능한 소프트웨어 취약성과 의도하지 않은 네트워크 노출이 있는지 모니터링하는 특수 설계된 스캔 엔진을 사용합니다. Amazon Inspector에서 소프트웨어 취약성 또는 의도하지 않은 네트워크 노출을 탐지하면 조사 결과가 생성됩니다. Amazon Inspector를 처음 활성화하면 Amazon EC2 스캔, Amazon ECR 스캔, Lambda 표준 스캔을 포함한 모든 스캔 유형에 계정이 자동으로 등록됩니다.

> **Note**
>
> Lambda 코드 스캔은 Lambda 함수 스캔의 선택적 계층으로, 언제든지 활성화할 수 있습니다.

주제

- Amazon Inspector 스캔 유형 개요
- 스캔 유형 활성화
- Amazon Inspector로 Amazon EC2 인스턴스 스캔
- Amazon Inspector로 Amazon Elastic Container Registry 컨테이너 이미지 스캔
- Amazon Inspector를 사용하여 AWS Lambda 함수 스캔
- Amazon Inspector에서 스캔 유형 비활성화

Amazon Inspector 스캔 유형 개요

Amazon Inspector는 AWS 환경의 특정 리소스 유형에 초점을 맞춘 다양한 스캔 유형을 제공합니다.

Amazon EC2 스캔

Amazon EC2 스캔을 활성화하면 Amazon Inspector에서 Amazon EC2 인스턴스를 스캔하여 일반적인 취약성 및 노출(CVE), 네트워크 노출 문제, 네트워크 연결성 문제, 운영 체제 및 프로그래밍 언어 패키지 취약성을 검사합니다. Amazon Inspector는 인스턴스에 설치된 SSM 에이전트를 사용하거나 인스턴스의 Amazon EBS 스냅샷을 통해 스캔을 수행합니다. 자세한 내용은 Amazon Inspector로 Amazon EC2 인스턴스 스캔 단원을 참조하십시오. 기본적으로 Amazon EC2 스캔을 활성화하면 하이브리드 스캔 모드가 자동으로 활성화됩니다. 자세한 내용은 에이전트 없는 스캔을 참조하세요.

Amazon ECR 스캔

Amazon ECR 스캔을 활성화하면 Amazon Inspector에서는 프라이빗 레지스트리의 모든 리포지토리를 기본 스캔 컨테이너 리포지토리에서 고급 스캔 리포지토리로 변환합니다. 푸시할 때만 스캔하거나 일부 리포지토리를 스캔하도록 포함 규칙을 통해 이 설정을 구성할 수 있습니다. Amazon Inspector는 ECR에서 활성 상태(`imageStatus` 필드가 임`ACTIVE`)인 ECR 컨테이너 이미지만 스캔합니다. Amazon Inspector는 지난 30일 이내에 ECR에서 활성(`lastActivatedAt`)으로 푸시 또는 전환되거나 지난 90일 이내에 가져온 모든 이미지를 스캔합니다. Amazon Inspector는 기본적으로 90일 동안 이미지를 계속 모니터링합니다. 이 설정은 언제든지 변경할 수 있습니다. 자세한 내용은 Amazon Inspector로 Amazon Elastic Container Registry 컨테이너 이미지 스캔 단원을 참조하십시오.

Lambda 표준 스캔

Lambda 표준 스캔을 활성화하면 Amazon Inspector는 계정에서 모든 Lambda 함수를 검색하고 즉시 취약성을 스캔합니다. Amazon Inspector는 배포 시 새 Lambda 함수 및 계층을 스캔합니다. Amazon Inspector는 업데이트되거나 새 CVE가 게시될 때 다시 스캔합니다. 자세한 내용은 Amazon Inspector를 사용하여 AWS Lambda 함수 스캔 섹션을 참조하세요.

Lambda 표준 스캔 + Lambda 코드 스캔

Lambda 코드 스캔을 활성화하면 Amazon Inspector는 계정에서 Lambda 함수 및 계층을 검색하고 코드 취약성을 스캔합니다. 이러한 유형의 스캔은 CVE에 대한 Lambda 함수에 사용되는 애플리케이션 패키지 종속성을 평가합니다. 이 스캔 유형을 활성화하면 Lambda 표준 스캔도 활성화됩니다. 자세한 내용은 Amazon Inspector를 사용하여 AWS Lambda 함수 스캔 단원을 참조하십시오.

Amazon Inspector용 코드 보안

이 스캔 유형은 Amazon Q Developer 스캔 엔진을 활용하여 자사 애플리케이션 코드, 타사 애플리케이션 종속성 및 취약성에 대한 코드형 인프라를 스캔합니다. 자세한 내용은 Amazon Inspector용 코드 보안을 참조하세요.

스캔 유형 활성화

스캔 유형은 언제든지 활성화할 수 있습니다. 스캔 유형을 활성화하면 Amazon Inspector에서 해당 스캔 유형의 적격 리소스를 스캔하기 시작합니다.

Amazon EC2 스캔

이 스캔 유형은 보안 권고에서 수집한 규칙과 메타데이터를 비교하기 전에 Amazon EC2 인스턴스에서 메타데이터를 추출합니다. 이 스캔 유형을 활성화하면 Amazon Inspector에서 계정의 모든 적격

Amazon EC2 인스턴스에 패키지 취약성 및 네트워크 연결성 문제가 있는지 검사합니다. 이 스캔 유형을 활성화한 후 인스턴스 탭에서 스캔 중인 인스턴스 수를 볼 수 있습니다.

Amazon ECR 스캔

이 스캔 유형은 Amazon ECR에서 컨테이너 이미지 및 컨테이너 리포지토리를 스캔합니다. 이 스캔 유형을 활성화하면 프라이빗 레지스트리의 스캔 구성 설정이 기본 스캔에서 고급 스캔으로 변경됩니다. Amazon ECR 스캔을 활성화한 후 컨테이너 이미지 및 컨테이너 리포지토리 탭에서 스캔 중인 이미지 및 리포지토리 수를 볼 수 있습니다.

Lambda 표준 스캔+ Lambda 코드 스캔

Lambda 표준 스캔은 기본 Lambda 스캔 유형입니다. Lambda 표준 스캔을 활성화하면 지난 90일 동안 간접적으로 호출되거나 업데이트된 적이 있는 모든 Lambda 함수에 대해 소프트웨어 취약성이 있는지 검사합니다. Lambda 표준 스캔을 활성화하면 Lambda 함수 탭에서 스캔 중인 Lambda 함수 수를 볼 수 있습니다.

Lambda 코드 스캔은 Lambda 함수에서 사용자 지정 애플리케이션 코드를 스캔합니다. Lambda 코드 스캔을 활성화하면 지난 90일 동안 간접적으로 호출되거나 업데이트된 적이 있는 모든 Lambda 함수에 대해 코드 취약성이 있는지 검사합니다. Lambda 표준 스캔을 활성화한 후 Lambda 함수 탭에서 코드 취약성에 대해 스캔 중인 Lambda 함수 수를 확인할 수 있습니다.

> **Note**
>
> Lambda 코드 스캔을 활성화하려면 먼저 Lambda 표준 스캔을 활성화해야 합니다.

Amazon Inspector 코드 보안

이 스캔 유형은 자사 애플리케이션 코드, 타사 애플리케이션 종속성 및 코드형 인프라에서 취약성을 스캔합니다. 코드 보안을 활성화하면 Amazon Inspector는 스캔 구성에 따라 코드 리포지토리에서 코드 취약성 검사를 시작합니다. Amazon Inspector 코드 보안을 활성화한 후 코드 리포지토리 탭에서 스캔 중인 코드 리포지토리 수를 볼 수 있습니다.

스캔 활성화

다음 절차에서는 Amazon Inspector에서 스캔 유형을 활성화하는 방법을 설명합니다.

> **Note**
>
> AWS 조직의 위임된 관리자인 경우 쉘 스크립트를 사용하여 여러 리전의 여러 계정에 대해 Amazon Inspector 스캔 유형을 활성화할 수 있습니다. 자세한 내용은 GitHub에서 inspector2-enablement-with-cli를 참조하세요. 그렇지 않으면 Amazon Inspector 위임된 관리자로 로그인한 상태에서 다음 단계를 완료합니다.

Console

스캔을 활성화하려면

1. Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 페이지 오른쪽 상단의 AWS 리전 선택기를 사용하여 새 스캔 유형을 활성화할 리전을 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다.
4. 계정 관리 페이지에서 스캔 유형을 활성화할 계정을 선택합니다.
5. 활성화를 선택하고 활성화하려는 스캔 유형을 선택합니다.
6. (권장) 해당 스캔 유형을 활성화 AWS 리전 하려는 각에서이 단계를 반복합니다.

API

Enable API 작업을 실행합니다. 요청에 스캔을 활성화할 계정 ID를 제공하고, 멱등성 토큰 및 `resourceTypes`로 `EC2`, `ECR`, `LAMBDA`, `LAMBDA_CODE` 중 하나 이상을 제공하면 해당 유형의 스캔이 활성화됩니다.

Amazon Inspector로 Amazon EC2 인스턴스 스캔

Amazon Inspector Amazon EC2 스캔은 보안 권고에서 수집한 규칙과 메타데이터를 비교하기 전에 EC2 인스턴스에서 메타데이터를 추출합니다. Amazon Inspector에서는 인스턴스에 패키지 취약성 및 네트워크 연결성 문제가 있는지 스캔하여 조사 결과를 생성합니다. Amazon Inspector는 12시간마다 한 번씩 네트워크 연결성 스캔을 수행하고 EC2 인스턴스와 연결된 스캔 방법에 따라 달라지는 가변 주기로 패키지 취약성 스캔을 수행합니다.

패키지 취약성 스캔은 에이전트 기반 또는 에이전트 없음 스캔 방법을 사용하여 수행할 수 있습니다. 이 두 가지 스캔 방법 모두 Amazon Inspector가 패키지 취약성 스캔을 위해 EC2 인스턴스에서 소프트

웨어 인벤토리를 수집하는 방법과 시기를 결정합니다. 에이전트 기반 스캔은 SSM 에이전트를 사용하여 소프트웨어 인벤토리를 수집하고, 에이전트 없는 스캔은 Amazon EBS 스냅샷을 사용하여 소프트웨어 인벤토리를 수집합니다.

Amazon Inspector는 계정에 대해 활성화한 스캔 방법을 사용합니다. Amazon Inspector를 처음 활성화하면 두 가지 스캔 방법을 모두 사용하는 하이브리드 스캔에 계정이 자동으로 등록됩니다. 하지만 언제든지 이 설정은 변경할 수 있습니다. 스캔 유형을 활성화하는 자세한 방법은 스캔 유형 활성화를 참조하세요. 이 단원에서는 Amazon EC2 스캔에 대한 정보를 제공합니다.

> **Note**
>
> Amazon EC2 스캔은 심층 검사를 통해 프로비저닝된 경우에도 가상 환경과 관련된 파일 시스템 디렉터리를 스캔하지 않습니다. 예를 들어 `/var/lib/docker/` 경로는 컨테이너 실행 시간에 일반적으로 사용되므로 스캔되지 않습니다.

에이전트 기반 스캔

에이전트 기반 스캔은 모든 적격 인스턴스에서 SSM 에이전트를 사용하여 연속으로 수행됩니다. 에이전트 기반 스캔의 경우 Amazon Inspector는 SSM 연결 및 이러한 연결을 통해 설치된 플러그인을 사용하여 인스턴스에서 소프트웨어 인벤토리를 수집합니다. Amazon Inspector 에이전트 기반 검사는 운영 체제 패키지에 대한 패키지 취약성 스캔뿐 아니라 Linux 기반 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사를 통해 Linux 기반 인스턴스의 애플리케이션 프로그래밍 언어 패키지에 대한 패키지 취약성 탐지도 가능합니다.

다음 프로세스는 Amazon Inspector에서 SSM을 사용하여 인벤토리를 수집하고 에이전트 기반 스캔을 수행하는 방법을 설명합니다.

1. Amazon Inspector는 계정에 SSM 연결을 생성하여 인스턴스에서 인벤토리를 수집합니다. 일부 인스턴스 유형(Windows 및 Linux)의 경우 이러한 연결이 개별 인스턴스에 플러그인을 설치하여 인벤토리를 수집합니다.
2. Amazon Inspector는 SSM을 사용하여 인스턴스에서 패키지 인벤토리를 추출합니다.
3. Amazon Inspector는 추출된 인벤토리를 평가하고 탐지된 취약성에 대한 조사 결과를 생성합니다.

> **Note**
>
> 에이전트 기반 스캔의 경우 동일한 AWS 계정의 SSM에서 Amazon EC2 인스턴스를 관리해야 합니다.

적격 인스턴스

Amazon Inspector는 다음 조건을 충족하는 경우 에이전트 기반 방법을 사용하여 인스턴스를 스캔합니다.

- 인스턴스에는 지원되는 OS가 있습니다. 지원되는 OS 목록은 the section called “지원되는 운영 체제: Amazon EC2 스캔”의 에이전트 기반 스캔 지원 열을 참조하세요.
- Amazon Inspector EC2 제외 태그로 인해 스캔에서 인스턴스가 제외되지 않습니다.
- 인스턴스가 SSM 관리형입니다. 에이전트를 확인하고 구성하는 방법은 SSM 에이전트 구성 섹션을 참조하세요.

에이전트 기반 스캔 동작

에이전트 기반 스캔 방법을 사용할 때 Amazon Inspector는 다음과 같은 경우에 EC2 인스턴스에 대한 새로운 취약성 스캔을 시작합니다.

- 새 EC2 인스턴스를 시작하는 경우
- 기존 EC2 인스턴스에 새 소프트웨어를 설치하는 경우(Linux 및 Mac)
- Amazon Inspector가 새로운 일반적인 취약성 및 노출(CVE) 항목을 데이터베이스에 추가하고, 해당 CVE가 EC2 인스턴스와 관련이 있는 경우(Linux 및 Mac)

Amazon Inspector는 초기 스캔이 완료되면 EC2 인스턴스의 마지막 스캔 필드를 업데이트합니다. 이후 Amazon Inspector가 SSM 인벤토리를 평가할 때(기본적으로 30분마다) 또는 해당 인스턴스에 영향을 미치는 새 CVE가 Amazon Inspector 데이터베이스에 추가되어 인스턴스를 다시 스캔할 때 마지막 스캔 필드가 업데이트됩니다.

EC2 인스턴스의 취약성을 마지막으로 스캔한 시기는 계정 관리 페이지의 인스턴스 탭에서 또는 ListCoverage 명령을 사용하여 확인할 수 있습니다.

SSM 에이전트 구성

Amazon Inspector에서 에이전트 기반 스캔 방법을 사용하여 Amazon EC2 인스턴스의 소프트웨어 취약성을 탐지하려면 해당 인스턴스가 Amazon EC2 Systems Manager(SSM)의 관리형 인스턴스여야 합니다. SSM 관리형 인스턴스에는 SSM 에이전트가 설치되어 실행 중이며 SSM에는 인스턴스 관리 권한이 있습니다. 이미 SSM을 사용하여 인스턴스를 관리하고 있는 경우 에이전트 기반 스캔을 위해 다른 단계가 필요하지 않습니다.

SSM 에이전트는 일부 Amazon Machine Images(AMI)에서 생성된 EC2 인스턴스에 기본적으로 설치됩니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 SSM 에이전트 정보를 참조하세요. 하지만 설치되어 있더라도 SSM 에이전트를 수동으로 활성화하고 SSM에 인스턴스 관리 권한을 부여해야 할 수 있습니다.

다음 절차에서는 IAM 인스턴스 프로파일을 사용하여 Amazon EC2 인스턴스를 관리형 인스턴스로 구성하는 방법에 대해 설명합니다. 이 절차에는 AWS Systems Manager 사용 설명서의 자세한 정보로 연결되는 링크도 제공됩니다.

AmazonSSMManagedInstanceCore는 인스턴스 프로파일을 연결할 때 권장되는 정책입니다. 이 정책에는 Amazon Inspector EC2 스캔에 필요한 모든 권한이 포함되어 있습니다.

> **Note**
>
> IAM 인스턴스 프로파일을 사용하지 않고 SSM 기본 호스트 관리 구성을 사용하여 모든 EC2 인스턴스의 SSM 관리를 자동화할 수도 있습니다. 자세한 내용은 기본 호스트 관리 구성을 참조하세요.

Amazon EC2 인스턴스에 대해 SSM을 구성하려면

1. 운영 체제 공급업체에서 아직 설치하지 않은 경우 SSM 에이전트를 설치합니다. 자세한 내용은 SSM 에이전트 작업을 참조하세요.
2. AWS CLI 를 사용하여 SSM 에이전트가 실행 중인지 확인합니다. 자세한 내용은 SSM 에이전트 상태 확인 및 에이전트 시작을 참조하세요.
3. SSM에 인스턴스 관리 권한을 부여합니다. IAM 인스턴스 프로파일을 생성한 후 이를 인스턴스에 연결하여 권한을 부여할 수 있습니다. Amazon Inspector에서 스캔하는 데 필요한 SSM Distributor, SSM Inventory 및 SSM State Manager에 대한 권한이 포함되어 있는 AmazonSSMManagedInstanceCore 정책을 사용하는 것이 좋습니다. 이러한 권한으로 인스턴스 프로파일을 생성하고 인스턴스에 연결하는 방법에 대한 지침은 Systems Manager에 대한 인스턴스 권한 구성을 참조하세요.

4. (선택 사항) SSM 에이전트에 대한 자동 업데이트를 활성화합니다. 자세한 내용은 SSM 에이전트 업데이트 자동화를 참조하세요.
5. (선택 사항) Amazon Virtual Private Cloud(Amazon VPC) 엔드포인트를 사용하도록 Systems Manager를 구성합니다. 자세한 내용은 Amazon VPC 엔드포인트 생성을 참조하세요.

> **Important**
>
> Amazon Inspector에서 소프트웨어 애플리케이션 인벤토리를 수집하려면 계정에 Systems Manager State Manager 연결이 필요합니다. 해당 연결이 없는 경우 Amazon Inspector에서 자동으로 `InspectorInventoryCollection-do-not-delete`라는 연결을 생성합니다. 또한 리소스 데이터 동기화도 필요하며, 아직 없는 경우 Amazon Inspector에서 자동으로 `InspectorResourceDataSync-do-not-delete`라는 동기화를 생성합니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 인벤토리의 리소스 데이터 동기화 구성을 참조하세요. 각 계정에는 리전당 리소스 데이터 동기화 수를 설정할 수 있습니다. 자세한 내용은 SSM 엔드포인트 및 할당량의 최대 리소스 데이터 동기화 수(리전 AWS 계정 당)를 참조하세요. https://docs.aws.amazon.com/general/latest/gr/ssm.html#limits_ssm

스캔을 위해 생성된 SSM 리소스

Amazon Inspector에서 Amazon EC2 스캔을 실행하려면 계정에 여러 SSM 리소스가 필요합니다. Amazon Inspector EC2 스캔을 처음 활성화하면 다음과 같은 리소스가 생성됩니다.

> **Note**
>
> 계정에 대해 Amazon Inspector Amazon EC2 스캔이 활성화되어 있는 동안 이러한 SSM 리소스가 삭제되는 경우, Amazon Inspector는 다음 스캔 간격에 해당 리소스를 다시 생성하려고 시도합니다.

`InspectorInventoryCollection-do-not-delete`

Amazon Inspector가 Amazon EC2 인스턴스에서 소프트웨어 애플리케이션 인벤토리를 수집할 때 사용하는 Systems Manager State Manager(SSM) 연결입니다. `InstanceIds*`에서 인벤토리를 수집하기 위한 SSM 연결이 계정에 이미 있는 경우 Amazon Inspector에서 자체적으로 생성하는 대신 해당 SSM 연결이 사용됩니다.

`InspectorResourceDataSync-do-not-delete`

Amazon Inspector가 Amazon EC2 인스턴스에서 수집된 인벤토리 데이터를 Amazon Inspector가 소유한 Amazon S3 버킷으로 보낼 때 사용하는 리소스 데이터 동기화입니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 인벤토리의 리소스 데이터 동기화 구성을 참조하세요.

`InspectorDistributor-do-not-delete`

Amazon Inspector에서 Windows 인스턴스를 스캔하는 데 사용하는 SSM 연결입니다. 이 연결은 Amazon Inspector SSM 플러그인을 Windows 인스턴스에 설치합니다. 플러그인 파일이 실수로 삭제된 경우 이 연결을 통해 다음 연결 간격에 다시 설치됩니다.

`InvokeInspectorSsmPlugin-do-not-delete`

Amazon Inspector에서 Windows 인스턴스를 스캔하는 데 사용하는 SSM 연결입니다. 이 연결을 통해 Amazon Inspector에서 플러그인을 사용하여 스캔을 시작할 수 있으며, 사용자도 이 연결을 통해 Windows 인스턴스 스캔에 대한 사용자 지정 간격을 설정할 수 있습니다. 자세한 내용은 Windows 인스턴스 스캔을 위한 사용자 지정 일정 설정 단원을 참조하십시오.

`InspectorLinuxDistributor-do-not-delete`

Amazon Inspector에서 Amazon EC2 Linux 심층 검사에 사용하는 SSM 연결입니다. 이 연결은 Amazon Inspector SSM 플러그인을 Linux 인스턴스에 설치합니다.

`InvokeInspectorLinuxSsmPlugin-do-not-delete`

Amazon Inspector에서 Amazon EC2 Linux 심층 검사에 사용하는 SSM 연결입니다. 이 연결을 통해 Amazon Inspector에서 플러그인을 사용하여 스캔을 시작할 수 있습니다.

> **Note**
>
> Amazon Inspector Amazon EC2 스캔 또는 심층 검사를 비활성화하면 SSM 리소스 `InvokeInspectorLinuxSsmPlugin-do-not-delete`가 더 이상 간접적으로 호출되지 않습니다.

에이전트 없는 스캔

Amazon Inspector는 계정이 하이브리드 스캔 모드인 경우 적격 인스턴스에 대해 에이전트 없는 스캔 방법을 사용합니다. 하이브리드 스캔 모드는 에이전트 기반 스캔과 에이전트 없는 스캔을 포함하며 Amazon EC2 스캔을 활성화하면 자동으로 활성화됩니다.

에이전트 없는 스캔의 경우 Amazon Inspector는 EBS 스냅샷을 사용하여 인스턴스에서 소프트웨어 인벤토리를 수집합니다. 에이전트 없는 스캔은 인스턴스에 운영 체제 및 애플리케이션 프로그래밍 언어 패키지 취약성이 있는지 스캔합니다.

> **Note**
>
> Linux 인스턴스에서 애플리케이션 프로그래밍 언어 패키지 취약성을 스캔할 때 에이전트 없는 스캔 방법은 사용 가능한 모든 경로를 검사하는 반면 에이전트 기반 스캔은 기본 경로와 사용자가 Linux 기반 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사의 일부로 지정한 추가 경로만 검사합니다. 이로 인해 에이전트 기반 방법을 사용하여 스캔했는지 아니면 에이전트 없는 스캔 방법을 사용하여 스캔했는지에 따라 동일한 인스턴스의 조사 결과가 달라질 수 있습니다.

다음 프로세스는 Amazon Inspector에서 EBS 스냅샷을 사용하여 인벤토리를 수집하고 에이전트 없는 스캔을 수행하는 방법을 설명합니다.

1. Amazon Inspector는 인스턴스에 연결된 모든 볼륨의 EBS 스냅샷을 생성합니다. Amazon Inspector에서 스냅샷을 사용하는 동안에는 스냅샷이 계정에 저장되고 `InspectorScan` 태그 키로 태그가 지정되고 고유한 스캔 ID가 태그 값으로 지정됩니다.
2. Amazon Inspector는 EBS 다이렉트 API를 사용하여 스냅샷에서 데이터를 검색하고 취약성이 있는지 평가합니다. 탐지된 모든 취약성에 대한 조사 결과가 생성됩니다.
3. Amazon Inspector는 계정에 생성한 EBS 스냅샷을 삭제합니다.

적격 인스턴스

Amazon Inspector는 다음 조건을 충족하는 경우 에이전트 없는 스캔 방법을 사용하여 인스턴스를 스캔합니다.

- 인스턴스에는 지원되는 OS가 있습니다. 자세한 내용은 the section called “지원되는 운영 체제: Amazon EC2 스캔”의 에이전트 기반 스캔 지원 열을 참조하세요.
- 인스턴스의 상태가 `Unmanaged EC2 instance`, `Stale inventory` 또는 `No inventory`입니다.
- 인스턴스가 Amazon EBS 기반이며 다음 파일 시스템 형식 중 하나를 사용합니다.
  - `ext3`
  - `ext4`

  - `xfs`
- Amazon EC2 제외 태그를 통해 인스턴스가 스캔에서 제외되지 않았습니다.
- 인스턴스에 연결된 볼륨의 수가 8개 미만이고 합산 크기가 1200GB 이하입니다.

에이전트 없는 스캔 동작

계정이 하이브리드 스캔을 사용하도록 구성된 경우 Amazon Inspector는 24시간마다 적격 인스턴스에 대해 에이전트 없는 스캔을 수행합니다. Amazon Inspector는 새로운 적격 인스턴스를 매시간 탐지하고 스캔하며, 여기에는 SSM 에이전트가 없는 새 인스턴스 또는 상태가 `SSM_UNMANAGED`로 변경된 기존 인스턴스가 포함됩니다.

Amazon Inspector는 에이전트 없는 스캔 후 인스턴스에서 추출된 스냅샷을 스캔할 때마다 Amazon EC2 인스턴스의 마지막 스캔 필드를 업데이트합니다.

EC2 인스턴스의 취약성을 마지막으로 스캔한 시기는 계정 관리 페이지의 인스턴스 탭에서 또는 ListCoverage 명령을 사용하여 확인할 수 있습니다.

스캔 모드 관리

EC2 스캔 모드는 Amazon Inspector가 계정에서 EC2 스캔을 수행할 때 사용할 스캔 방법을 결정합니다. 일반 설정의 EC2 스캔 설정 페이지에서 계정의 스캔 모드를 볼 수 있습니다. 독립형 계정 또는 Amazon Inspector 위임 관리자는 스캔 모드를 변경할 수 있습니다. Amazon Inspector 위임 관리자 권한으로 스캔 모드를 설정하면 조직의 모든 멤버 계정에 해당 스캔 모드가 설정됩니다. Amazon Inspector에는 다음과 같은 스캔 모드가 있습니다.

에이전트 기반 스캔 - 이 스캔 모드에서는 Amazon Inspector가 패키지 취약성을 검사할 때 에이전트 기반 스캔 방법만 사용합니다. 이 스캔 모드는 계정의 SSM 관리형 인스턴스만 스캔하지만 새로운 CVE 또는 인스턴스 변경에 대한 응답으로 연속 스캔을 제공한다는 이점이 있습니다. 에이전트 기반 스캔은 Amazon Inspector에 적격 인스턴스에 대한 심층 검사도 제공합니다. 새로 활성화된 계정에서는 이 모드가 기본 스캔 모드입니다.

하이브리드 스캔 - 이 스캔 모드에서 Amazon Inspector는 에이전트 기반과 에이전트 없음 스캔 방법을 조합하여 패키지 취약성을 스캔합니다. SSM 에이전트가 설치 및 구성된 적격 EC2 인스턴스의 경우 Amazon Inspector는 에이전트 기반 방법을 사용합니다. SSM 관리형이 아닌 적격 인스턴스의 경우 Amazon Inspector는 EBS 지원 적격 인스턴스에 에이전트 없는 스캔 방법을 사용합니다.

스캔 모드를 변경하는 방법

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 페이지 오른쪽 상단의 AWS 리전 선택기를 사용하여 EC2 스캔 모드를 변경할 리전을 선택합니다.
3. 측면 탐색 패널의 일반 설정에서 EC2 스캔 설정을 선택합니다.
4. 스캔 모드에서 편집을 선택합니다.
5. 스캔 모드를 선택한 다음 변경 사항 저장을 선택합니다.

Amazon Inspector 스캔에서 인스턴스 제외

Linux 및 Windows 인스턴스에 `InspectorEc2Exclusion` 키로 태그를 지정하여 Amazon Inspector 스캔에서 해당 인스턴스를 제외할 수 있습니다. 태그 키는 대/소문자를 구분하지 않습니다. 태그 값을 포함하는 것은 선택 사항입니다. 태그 추가에 대한 자세한 내용은 Amazon EC2 리소스 태깅을 참조하세요.

Amazon Inspector 스캔에서 제외하도록 인스턴스에 태그를 지정하면 Amazon Inspector는 해당 인스턴스를 제외된 것으로 표시하고 조사 결과를 생성하지 않습니다. 그러나 Amazon Inspector SSM 플러그인은 계속 간접적으로 호출됩니다. 플러그인이 간접적으로 호출되지 않도록 하려면 인스턴스 메타데이터의 태그에 대한 액세스를 허용해야 합니다.

> **Note**
>
> 제외된 인스턴스에 대해서는 요금이 부과되지 않습니다.

또한 해당 볼륨을 암호화하는 데 사용되는 AWS KMS 키에 태그를 지정하여 에이전트 없는 스캔에서 암호화된 EBS 볼륨을 제외할 수 있습니다`InspectorEc2Exclusion`. 자세한 내용은 키 태그 지정을 참조하세요.

지원되는 운영 체제

Amazon Inspector는 지원되는 Mac, Windows 및 Linux 인스턴스에서 운영 체제 패키지의 취약성을 스캔합니다. Linux 인스턴스의 경우 Amazon Inspector는 Linux 기반 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사를 사용하여 애플리케이션 프로그래밍 언어 패키지에 대한 조사 결과를 생성할 수 있습니다. Mac 및 Windows 인스턴스의 경우 운영 체제 패키지만 스캔됩니다.

SSM 에이전트 없이 스캔할 수 있는 운영 체제를 포함하여 지원되는 운영 체제에 대한 자세한 내용은 섹션을 참조하세요Amazon EC2 인스턴스 상태 값.

Linux 기반 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사

Amazon Inspector의 Amazon EC2 스캔 적용 범위가 확대되어 심층 검사가 포함되었습니다. 심층 검사를 통해 Amazon Inspector는 Linux 기반 Amazon EC2 인스턴스에서 애플리케이션 프로그래밍 언어 패키지의 패키지 취약성을 탐지합니다. Amazon Inspector는 프로그래밍 언어 패키지 라이브러리의 기본 경로를 스캔합니다. 그러나 Amazon Inspector에서 기본적으로 스캔하는 경로 외에 사용자 지정 경로를 구성할 수 있습니다.

> **Note**
>
> 심층 검사는 기본 호스트 관리 구성 설정으로 사용할 수 있습니다. 그러나 `ssm:PutInventory` 및 `ssm:GetParameter` 권한으로 구성된 역할을 생성하거나 사용해야 합니다.

Linux 기반 Amazon EC2 인스턴스에 대한 심층 검사 스캔을 수행하기 위해 Amazon Inspector는 Amazon Inspector SSM 플러그인으로 수집한 데이터를 사용합니다. Amazon Inspector SSM 플러그인을 관리하고 Linux에 대한 심층 검사를 수행하기 위해 Amazon Inspector는 계정에 SSM 연결 `InvokeInspectorLinuxSsmPlugin-do-not-delete`를 자동으로 생성합니다. Amazon Inspector는 Linux 기반 Amazon EC2 인스턴스에서 6시간마다 업데이트된 애플리케이션 인벤토리를 수집합니다.

> **Note**
>
> Windows 또는 Mac 인스턴스에는 심층 검사가 지원되지 않습니다.

이 단원에서는 Amazon Inspector가 스캔할 사용자 지정 경로를 설정하는 방법을 포함하여 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사를 관리하는 방법에 대해 설명합니다.

주제

- 심층 검사 액세스 또는 비활성화
- Amazon Inspector 심층 검사를 위한 사용자 지정 경로
- Amazon Inspector 심층 검사를 위한 사용자 지정 일정

- 지원되는 프로그래밍 언어

심층 검사 액세스 또는 비활성화

> **Note**
>
> 2023년 4월 17일 이후에 Amazon Inspector를 활성화하는 계정의 경우, Amazon EC2 스캔의 일부로 심층 검사가 자동으로 활성화됩니다.

심층 검사를 관리하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 일반 설정을 선택한 다음 Amazon EC2 스캔 설정을 선택합니다.
3. Amazon EC2 인스턴스 심층 검사에서 조직 또는 자신의 계정에 대한 사용자 지정 경로를 설정할 수 있습니다.

단일 계정에 대한 활성화 상태는 GetEc2DeepInspectionConfiguration API를 사용하여 프로그래밍 방식으로 확인할 수 있습니다. 여러 계정에 대한 활성화 상태는 BatchGetMemberEc2DeepInspectionStatus API를 사용하여 프로그래밍 방식으로 확인할 수 있습니다.

2023년 4월 17일 이전에 Amazon Inspector를 활성화한 경우 콘솔 배너 또는 UpdateEc2DeepInspectionConfiguration API를 통해 심층 검사를 활성화할 수 있습니다. Amazon Inspector에서 조직의 위임된 관리자인 경우 BatchUpdateMemberEc2DeepInspectionStatus API를 사용하여 자신과 멤버 계정에 대한 심층 검사를 활성화할 수 있습니다.

UpdateEc2DeepInspectionConfiguration API를 통해 심층 검사를 비활성화할 수 있습니다. 조직의 멤버 계정으로는 심층 검사를 비활성화할 수 없습니다. 대신 위임 관리자가 BatchUpdateMemberEc2DeepInspectionStatus API를 사용하여 멤버 계정을 비활성화해야 합니다.

Amazon Inspector 심층 검사를 위한 사용자 지정 경로

Linux Amazon EC2 인스턴스의 심층 검사 중에 Amazon Inspector가 스캔할 사용자 지정 경로를 설정할 수 있습니다. 사용자 지정 경로를 설정하면 Amazon Inspector는 해당 디렉터리와 그 안의 모든 하위 디렉터리에 있는 패키지를 스캔합니다.

모든 계정은 최대 5개까지 사용자 지정 경로를 정의할 수 있습니다. 조직의 위임된 관리자는 사용자 지정 경로를 10개 정의할 수 있습니다.

Amazon Inspector는 모든 계정을 대상으로 스캔하는 Amazon Inspector의 다음 기본 경로 외에 모든 사용자 지정 경로를 스캔합니다.

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

> **Note**
>
> 사용자 지정 경로는 로컬 경로여야 합니다. Amazon Inspector는 네트워크 파일 시스템 마운트 또는 Amazon S3 파일 시스템 마운트와 같은 매핑된 네트워크 경로를 스캔하지 않습니다.

사용자 지정 경로의 형식

사용자 지정 경로는 256자를 초과할 수 없습니다. 다음은 사용자 지정 경로가 어떻게 표시되는지 보여주는 예시입니다.

경로 예

`/home/usr1/project01`

> **Note**
>
> 인스턴스당 패키지 한도는 5,000입니다. 최대 패키지 인벤토리 수집 시간은 15분입니다. Amazon Inspector에서는 이러한 제한을 피하기 위해 사용자 지정 경로를 선택할 것을 권장합니다.

Amazon Inspector 콘솔 및 Amazon Inspector API에서 사용자 지정 경로 설정

다음 절차에서는 Amazon Inspector 콘솔과 Amazon Inspector API를 사용하여 Amazon Inspector 심층 검사를 위한 사용자 지정 경로를 설정하는 방법에 대해 설명합니다. 사용자 지정 경로를 설정하면 Amazon Inspector에서 다음 정밀 검사에 해당 경로를 포함합니다.

Console

1. 위임된 관리자 AWS Management Console 로에 로그인하고 https://console.aws.amazon.com/inspector/v2/home Amazon Inspector 콘솔을 엽니다.
2. AWS 리전 선택기를 사용하여 Lambda 표준 스캔을 활성화할 리전을 선택합니다.
3. 탐색 창에서 일반 설정을 선택한 다음 EC2 스캔 설정을 선택합니다.
4. 자체 계정의 사용자 지정 경로에서 편집을 선택합니다.
5. 경로 텍스트 상자에 사용자 지정 경로를 입력합니다.
6. 저장을 선택합니다.

API

UpdateEc2DeepInspectionConfiguration 명령을 실행합니다. `packagePaths`의 경우 스캔할 경로 배열을 지정합니다.

Amazon Inspector 심층 검사를 위한 사용자 지정 일정

기본적으로 Amazon Inspector는 6시간마다 Amazon EC2 인스턴스에서 애플리케이션 인벤토리를 수집합니다. 그러나 다음 명령을 실행하여 Amazon Inspector가 이 작업을 수행하는 빈도를 제어할 수 있습니다.

예제 명령 1: 연결 ID 및 현재 간격을 보기 위해 연결 나열

다음 명령은 연결 `InvokeInspectorLinuxSsmPlugin-do-not-delete`의 연결 ID를 보여줍니다.

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

예제 명령 2: 새 간격을 포함하도록 연결 업데이트

다음 명령은 연결 `InvokeInspectorLinuxSsmPlugin-do-not-delete`의 연결 ID를 사용합니다. `schedule-expression`의 속도를 6시간에서 새로운 간격(예: 12시간)으로 설정할 수 있습니다.

```
aws ssm update-association \
--association-id "your-association-ID" \
--association-name "InvokeInspectorLinuxSsmPlugin-do-not-delete" \
```

```
--schedule-expression "rate(6 hours)" \
--region your-Region
```

> **Note**
>
> 사용 사례에 따라 `schedule-expression`의 속도를 6시간에서 30분과 같은 간격으로 설정하면 일일 SSM 인벤토리 한도를 초과할 수 있습니다. 이로 인해 결과가 지연되고 부분 오류 상태의 Amazon EC2 인스턴스가 발생할 수 있습니다.

지원되는 프로그래밍 언어

Linux 인스턴스의 경우, Amazon Inspector 심층 검사를 통해 애플리케이션 프로그래밍 언어 패키지 및 운영 체제 패키지에 대한 조사 결과를 생성할 수 있습니다.

Mac 및 Windows 인스턴스의 경우, Amazon Inspector 심층 검사를 통해 운영 체제 패키지에 대한 조사 결과만 생성할 수 있습니다.

지원되는 프로그래밍 언어에 대한 자세한 내용은 지원되는 프로그래밍 언어: Amazon EC2 심층 검사를 참조하세요.

Amazon Inspector로 Windows EC2 인스턴스 스캔

Amazon Inspector는 지원되는 모든 Windows 인스턴스를 자동으로 검색하여 추가 조치 없이 연속 스캔에 포함시킵니다. 지원되는 인스턴스에 대한 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요. Amazon Inspector는 정기적으로 Windows 스캔을 실행합니다. Windows 인스턴스는 검색 시 스캔되고 그 후 6시간마다 스캔됩니다. 그러나 첫 번째 스캔 후 기본 스캔 간격을 조정할 수 있습니다.

Amazon EC2 스캔이 활성화되면 Amazon Inspector에서 Windows 리소스에 대해 새 SSM 연결 `InspectorDistributor-do-not-delete`, `InspectorInventoryCollection-do-not-delete`, `InvokeInspectorSsmPlugin-do-not-delete`를 생성합니다. Windows 인스턴스에 Amazon Inspector SSM 플러그인을 설치하기 위해 `InspectorDistributor-do-not-delete` SSM 연결에는 `AWS-ConfigureAWSPackage` SSM 문서와 `AmazonInspector2-InspectorSsmPlugin` SSM Distributor 패키지가 사용됩니다. 자세한 내용은 Windows용 Amazon Inspector SSM 플러그인을 참조하세요. 인스턴스 데이터를 수집하고 Amazon Inspector 조사 결과를 생성하기 위해 `InvokeInspectorSsmPlugin-do-not-delete` SSM 연결은 Amazon Inspector SSM 플러그인을 6시간 간격으로 실행합니다. 하지만 cron 또는 rate 표현식을 설정하여 이 설정을 사용자 지정할 수 있습니다.

> **Note**
>
> Amazon Inspector는 업데이트된 OVAL(Open Vulnerability and Assessment Language) 정의 파일을 S3 버킷 `inspector2-oval-prod-your-AWS-Region`에 스테이징합니다. Amazon S3 버킷에는 스캔에 사용되는 OVAL 정의가 포함되어 있습니다. 이러한 OVAL 정의는 수정해서는 안됩니다. 그렇지 않으면 새로운 CVE가 릴리스될 때 Amazon Inspector가 이를 스캔하지 못하게 됩니다.

Windows 인스턴스에 대한 Amazon Inspector 스캔 요구 사항

Amazon Inspector에서 Windows 인스턴스를 스캔하려면 인스턴스가 다음 기준을 충족해야 합니다.

- 인스턴스가 SSM 관리형 인스턴스입니다. 스캔할 인스턴스 설정에 대한 지침은 SSM 에이전트 구성 섹션을 참조하세요.
- 인스턴스 운영 체제가 지원되는 Windows 운영 체제 중 하나입니다. 지원되는 운영 체제의 전체 목록은 Amazon EC2 인스턴스 상태 값 섹션을 참조하세요.
- 인스턴스에 Amazon Inspector SSM 플러그인이 설치되어 있습니다. Amazon Inspector는 관리형 인스턴스가 검색되면 자동으로 Amazon Inspector SSM 플러그인을 설치합니다. 플러그인에 대한 자세한 내용은 다음 주제를 참조하세요.

> **Note**
>
> 호스트가 외부 인터넷에 액세스할 수 없는 Amazon VPC에서 실행되는 경우 Windows 스캔을 수행하려면 호스트가 리전 Amazon S3 엔드포인트에 액세스할 수 있어야 합니다. Amazon S3 Amazon VPC 엔드포인트를 구성하는 방법을 알아보려면 Amazon Virtual Private Cloud 사용 설명서에서 게이트웨이 엔드포인트 생성을 참조하세요. Amazon VPC 엔드포인트 정책이 외부 S3 버킷에 대한 액세스를 제한하는 경우 인스턴스를 평가하는 데 사용되는 OVAL 정의를 AWS 리전 저장하는에서 Amazon Inspector가 유지 관리하는 버킷에 대한 액세스를 구체적으로 허용해야 합니다. 이 버킷의 형식은 `inspector2-oval-prod-REGION`입니다.

Windows 인스턴스 스캔을 위한 사용자 지정 일정 설정

SSM을 통해 `InvokeInspectorSsmPlugin-do-not-delete` 연결에 대한 cron 표현식 또는 rate 표현식을 설정하여 Windows Amazon EC2 인스턴스의 스캔 간격을 사용자 지정할 수 있습니다. 자세

한 내용은 AWS Systems Manager 사용 설명서의 참조: Systems Manager의 Cron 및 Rate 표현식을 참조하거나 다음 지침을 따르세요.

다음 코드 예제 중 하나를 선택하여 rate 표현식 또는 cron 표현식을 사용하여 Windows 인스턴스의 스캔 주기를 기본 6시간에서 12시간으로 변경할 수 있습니다.

다음 예에서는 `InvokeInspectorSsmPlugin-do-not-delete` 연결에 AssociationId를 사용해야 합니다. 다음 AWS CLI 명령을 실행하면 AssociationID를 검색할 수 있습니다.

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

> **Note**
>
> AssociationId는 리전별이므로 먼저 각에 대한 고유 ID를 검색해야 합니다 AWS 리전. 그런 다음 명령을 실행하여 Windows 인스턴스에 대한 사용자 지정 스캔 일정을 설정하려는 각 리전의 스캔 주기를 변경할 수 있습니다.

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 * * ? *)"
```

Amazon Inspector로 Amazon Elastic Container Registry 컨테이너 이미지 스캔

Amazon Inspector는 Amazon Elastic Container Registry에 저장된 컨테이너 이미지에 소프트웨어 취약성이 있는지 스캔하여 패키지 취약성 조사 결과를 생성합니다. Amazon ECR 스캔을 활성화할 때 Amazon Inspector를 프라이빗 레지스트리에 대한 기본 스캔 서비스로 설정하세요.

> **Note**
>
> Amazon ECR은 레지스트리 정책을 사용하여 AWS 보안 주체에 권한을 부여합니다. 이 위탁자는 스캔을 위해 Amazon Inspector API를 직접적으로 호출하는 데 필요한 권한을 가집니다. 레지스트리 정책의 범위를 설정할 때 deny에서 `ecr:*` 작업 또는 `PutRegistryScanningConfiguration`을 추가해서는 안 됩니다. 이로 인해 Amazon ECR에 대한 스캔을 활성화 및 비활성화할 때 레지스트리 수준에서 오류가 발생합니다.

기본 스캔을 사용하면 푸시할 때 스캔하거나 수동 스캔을 수행하도록 리포지토리를 구성할 수 있습니다. 고급 스캔을 사용하면 레지스트리 수준에서 운영 체제 및 프로그래밍 언어 패키지 취약성을 스캔할 수 있습니다. 기본 스캔과 고급 스캔의 차이점을 나란히 비교하려면 Amazon Inspector FAQ를 참조하세요.

> **Note**
>
> 기본 스캔은 Amazon ECR을 통해 제공되며 요금이 청구됩니다. 자세한 내용은 Amazon Elastic Container Registry 요금을 참조하세요. 고급 스캔은 Amazon Inspector를 통해 제공되며 요금이 청구됩니다. 자세한 내용은 Amazon Inspector 요금을 참조하세요.

Amazon ECR 스캔을 활성화하는 자세한 방법은 스캔 유형 활성화를 참조하세요. 조사 결과를 보는 방법에 대한 자세한 내용은 Amazon Inspector 조사 결과 보기를 참조하세요. 이미지 수준에서 Amazon ECR 조사 결과를 보는 방법에 대한 자세한 내용은 Amazon Elastic Container Registry 사용 설명서에서 이미지 스캔을 참조하세요. AWS Security Hub CSPM 및 Amazon EventBridge와 같은 기본 스캔에 사용할 수 AWS 서비스 없는를 사용하여 조사 결과를 관리할 수 있습니다.

적용 범위 페이지 및 API를 통해 Amazon Inspector의 각 리포지토리에 대한 스캔 구성을 볼 수 있습니다. 그러나 기본 스캔과 연속 스캔의 구성 설정은 Amazon ECR에서만 수정할 수 있습니다. Amazon Inspector는 이러한 설정에 대한 가시성을 제공하지만 직접 수정 기능은 제공하지 않습니다. 자세한 내용은 Amazon ECR 사용 설명서의 이미지 스캔에서 Amazon ECR의 소프트웨어 취약성을 참조하세요.

이 단원에서는 Amazon ECR 스캔에 대한 정보를 제공하고 Amazon ECR 리포지토리에 대해 고급 스캔을 구성하는 방법에 대해 설명합니다.

Amazon ECR 스캔의 스캔 동작

Amazon ECR 스캔을 처음 활성화하면 Amazon Inspector는 지난 14일 이내에 푸시된 이미지를 스캔합니다. 그런 다음 Amazon Inspector는 이미지를 스캔하여 스캔 상태를 `ACTIVE`로 설정합니다. Amazon Inspector는 ECR에서 활성화된 이미지만 스캔합니다(`imageStatus` 필드는). `ACTIVE` 보관된 상태가 ECR(`imageStatus` 필드는 `ARCHIVED`)인 이미지는 Amazon Inspector에서 스캔하지 않습니다.

연속 스캔이 활성화된 경우 Amazon Inspector는 14일(기본값) 이내에 푸시되었거나 마지막 사용 날짜가 14일(기본값) 이내이거나, 구성된 재스캔 기간 내에 이미지가 스캔되는 한 이미지를 모니터링합니다. 2025년 5월 16일 이전에 생성된 Amazon Inspector 계정의 경우, 기본 구성은 지난 90일 이내에 푸시되거나 가져온 이미지를 다시 스캔하여 모니터링하는 것입니다. 자세한 내용은 Amazon ECR 재스캔 기간 구성을 참조하세요.

연속 스캔을 위해 Amazon Inspector는 다음과 같은 경우에 컨테이너 이미지의 새로운 취약성 스캔을 시작합니다.

- 새 컨테이너 이미지가 푸시될 때마다
- Amazon Inspector가 새로운 일반적인 취약성 및 노출(CVE) 항목을 데이터베이스에 추가하고, 해당 CVE가 해당 컨테이너 이미지와 관련이 있을 때마다(연속 스캔만 해당)
- 컨테이너 이미지가 ECR에서 아카이브됨에서 활성으로 전환될 때마다

푸시할 때 스캔하도록 저장소를 구성하면 이미지를 푸시할 때만 이미지가 스캔됩니다.

컨테이너 이미지의 취약성을 마지막으로 검사한 시기는 계정 관리 페이지의 컨테이너 이미지 탭에서 또는 ListCoverage API를 사용하여 확인할 수 있습니다. Amazon Inspector는 다음 이벤트에 대한 응답으로 Amazon ECR 이미지의 마지막 스캔 시간 필드를 업데이트합니다.

- Amazon Inspector에서 컨테이너 이미지의 첫 번째 스캔을 완료한 경우
- 컨테이너 이미지에 영향을 미치는 새로운 일반적인 취약성 및 노출(CVE) 항목이 Amazon Inspector 데이터베이스에 추가되어 Amazon Inspector에서 컨테이너 이미지를 다시 스캔하는 경우

보관된 ECR 컨테이너 이미지

Amazon Inspector는 ECR에 보관된 컨테이너 이미지를 스캔하지 않습니다(`imageStatus`는 `ARCHIVED`). ECR의 활성 이미지가 아카이브로 전환되면 Amazon Inspector는 자동으로 조사 결과를

닫은 다음 3일 후에 조사 결과를 삭제합니다. 보관된 컨테이너 이미지가 ECR에서 활성으로 전환되면 Amazon Inspector가 새 스캔을 트리거합니다.

실행 중인 컨테이너에 컨테이너 이미지 매핑

Amazon Inspector는 Amazon Elastic Container Service(Amazon ECS) 및 Amazon Elastic Kubernetes Service(Amazon EKS)에서 실행 중인 컨테이너에 컨테이너 이미지를 매핑하여 포괄적인 컨테이너 보안 관리를 제공합니다. 이러한 매핑은 실행 중인 컨테이너의 이미지 취약성에 대한 인사이트를 제공합니다.

> **Note**
>
> 관리형 정책 `AWSReadOnlyAccess`만으로는 Amazon ECR 이미지와 실행 중인 컨테이너 간의 매핑을 볼 수 있는 충분한 권한을 제공하지 않습니다. 컨테이너 이미지 매핑 정보를 보려면 `AWSReadOnlyAccess` 및 `AWSInspector2ReadOnlyAccess` 관리형 정책이 모두 필요합니다.

운영 위험을 기반으로 문제 해결 작업의 우선순위를 정하고 전체 컨테이너 에코시스템에서 보안 범위를 유지할 수 있습니다. 현재 사용 중인 컨테이너 이미지 수와 지난 24시간 동안 Amazon ECS 또는 Amazon EKS 클러스터에서 마지막으로 사용된 컨테이너 이미지를 볼 수 있습니다. 배포된 Amazon ECS 작업 및 Amazon EKS 포드 수를 볼 수도 있습니다. 이 정보는 컨테이너 이미지 조사 결과에 대한 세부 정보 화면의 Amazon Inspector 콘솔과 `FilterCriteria` 데이터 유형에 대한 `ecrImageInUseCount` 및 `ecrImageLastInUseAt` 필터에서 찾을 수 있습니다. 새 컨테이너 이미지 또는 계정의 경우 데이터를 사용할 수 있는 데 최대 36시간이 걸릴 수 있습니다. 그런 다음이 데이터는 24시간마다 한 번씩 업데이트됩니다. 자세한 내용은 Amazon Inspector 조사 결과 보기 및 Amazon Inspector 조사 결과에 대한 세부 정보 보기를 참조하세요.

> **Note**
>
> 이 데이터는 Amazon ECR 스캔을 활성화하고 연속 스캔을 위해 리포지토리를 구성할 때 Amazon ECR 조사 결과로 자동으로 전송됩니다. 연속 스캔은 Amazon ECR 리포지토리 수준에서 구성해야 합니다. 자세한 내용은 Amazon Elastic Container Registry 사용 설명서의 고급 스캔을 참조하세요.

마지막 사용 날짜를 기준으로 클러스터에서 컨테이너 이미지를 다시 스캔할 수도 있습니다.

이 특성은 Amazon ECS 및 Amazon EKS를 사용하는 Fargate에서도 지원됩니다.

지원되는 운영 체제 및 미디어 유형

지원되는 운영 체제에 대한 자세한 내용은 지원되는 운영 체제: Amazon Inspector를 사용한 Amazon ECR 스캔 섹션을 참조하세요.

Amazon Inspector의 Amazon ECR 리포지토리 스캔에서 지원되는 미디어 유형은 다음과 같습니다.

이미지 매니페스트

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

이미지 구성

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

이미지 계층

- `"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 Inspector는 Amazon ECR 리포지토리 스캔을 위한 `"application/vnd.docker.distribution.manifest.list.v2+json"` 미디어 유형을 지원하지 않습니다.

Amazon ECR 재스캔 기간 구성

Amazon ECR 재스캔 기간 설정에 따라 Amazon Inspector에서 리포지토리의 컨테이너 이미지를 지속적으로 모니터링하는 기간이 결정됩니다. 이미지 마지막 사용 날짜, 이미지 가져오기 날짜 및 푸시 날짜에 대한 재스캔 기간을 구성하세요. 가장 좋은 방법은 사용자 환경에 가장 적합하도록 재검색 기간을 구성하는 것입니다.

예를 들어 이미지를 자주 빌드하는 경우 더 짧은 스캔 기간을 선택합니다. 장기간에 걸쳐 사용하는 이미지의 경우 더 긴 스캔 기간을 선택합니다. 조직에 추가된 새 계정을 포함하여 새 계정의 기본 스캔 기간은 14일입니다.

Amazon Inspector는 클러스터에서 마지막으로 사용 중이거나 14일 이내에 푸시된(기본적으로) 이미지를 계속 모니터링하고 다시 스캔합니다. 구성된 푸시 및 마지막 사용 날짜 내에 이미지가 푸시되지 않았거나 실행 중인 컨테이너에서 마지막으로 사용되지 않은 경우 Amazon Inspector는 이미지 모니터링을 중지합니다. 필요한 경우 마지막 사용 날짜 대신 마지막 가져오기 날짜별로 이미지를 모니터링하도록 설정을 변경하는 옵션이 있습니다. Amazon Inspector에서 이미지 모니터링을 중지하면 이미지 스캔 상태 코드가 비활성으로 설정되고 사유 코드는 만료로 설정됩니다. 그런 다음 Amazon Inspector는 관련된 모든 이미지 조사 결과를 종료하도록 예약합니다.

푸시 날짜 기간을 늘리면 Amazon Inspector는 지속적인 스캔을 수행하도록 구성된 리포지토리의 모든 활성 스캔 이미지에 변경 사항을 적용합니다. 그러나 비활성 이미지는 새 기간 내에 푸시했더라도 비활성 상태로 유지됩니다.

위임된 관리자 계정에서 스캔 기간을 구성하면 Amazon Inspector는 조직의 모든 멤버 계정에 해당 설정을 적용합니다. 위임된 관리자 계정이 Amazon ECR 스캔을 활성화하지 않으면 API 이미지에 대한 클러스터를 볼 수 없습니다.

다중 아키텍처 이미지의 경우 마지막 사용 날짜 추적은 지원되지 않습니다. 다중 아키텍처 이미지를 사용하는 경우 적절한 재스캔 동작을 보장하기 위해 마지막 사용 날짜 대신 이미지 가져오기 또는 푸시 이벤트를 기반으로 스캔을 구성하는 것이 좋습니다.

> **Note**
>
> 2025년 5월 16일 이전에 구성된 모든 재스캔 기간 설정은 변경되지 않습니다. 이전에 구성한 기본 설정을 계속 사용할 수 있습니다.

이미지 재스캔 기간

이미지 재스캔 기간은 Amazon Inspector가 이미지를 모니터링하는 기간을 결정합니다. 이미지 재스캔 지속 시간에는 마지막 사용 날짜(기본값) 또는 마지막 가져오기 날짜의 두 가지 모드가 포함됩니다. Amazon ECS/Amazon EKS 클러스터 활동의 마지막 사용 날짜를 사용하려면 마지막 사용 날짜(기본값)를 선택합니다. Amazon ECR 이미지의 마지막 가져오기 날짜를 사용하여 이미지를 다시 스캔하려면 마지막 가져오기 날짜를 선택합니다. 다음 옵션을 재스캔 기간으로 사용할 수 있습니다.

- 14일(기본값)
- 30일
- 60일
- 90일
- 180일

이미지 푸시 날짜 기간

이미지 푸시 날짜 기간은 이미지가 리포지토리로 푸시된 후 Amazon Inspector에서 이미지를 지속적으로 모니터링하는 기간을 결정합니다. 다음 옵션을 재스캔 기간으로 사용할 수 있습니다.

- 14일(기본값)
- 30일
- 60일
- 90일
- 180일
- 수명

Amazon ECR 재스캔 기간을 구성하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. Amazon ECR 재스캔 기간을 구성할 AWS 리전 를 선택합니다.
3. 탐색 창에서 일반 설정을 선택한 다음 ECR 스캔 설정을 선택합니다.
4. ECR 재스캔 기간에서 이미지 재스캔 모드를 선택한 다음 해당 기간을 선택합니다.
5. 이미지 푸시 날짜에서 이미지 푸시 날짜를 선택합니다.
6. 저장을 선택합니다.

ECR 컨테이너 이미지 상태 이해

Inspector는 ECR 컨테이너 `ACTIVE` 이미지의 이미지만 스캔합니다. `ARCHIVED` 상태의 ECR 컨테이너 이미지는 스캔되지 않습니다. 스캔 동작에 대한 자세한 내용은 섹션을 참조하세요Amazon ECR 스캔의 스캔 동작.

ECR의 ECR 컨테이너 이미지 상태가 로 전환되면 `ACTIVE`Inspector는 `lastActivatedAt` 필드를 사용하여 재스캔 기간을 모니터링합니다.

Amazon Inspector를 사용하여 AWS Lambda 함수 스캔

AWS Lambda 함수 및 계층에 대한 Amazon Inspector 지원은 지속적인 자동 보안 취약성 평가를 제공합니다. Amazon Inspector는 두 가지 유형의 Lambda 함수 스캔을 제공합니다.

Amazon Inspector Lambda 표준 스캔

이 스캔 유형이 기본 Lambda 스캔 유형입니다. Lambda 함수 및 계층의 애플리케이션 종속성을 스캔하여 패키지 취약성을 찾아냅니다.

Amazon Inspector Lambda 코드 스캔

이 스캔 유형은 Lambda 함수 및 계층의 사용자 지정 애플리케이션 코드를 스캔하여 코드 취약성을 찾아냅니다. Lambda 표준 스캔을 활성화하거나, Lambda 표준 스캔을 Lambda 코드 스캔과 함께 활성화할 수 있습니다.

Lambda 코드 스캔을 활성화하려면 먼저 Lambda 표준 스캔을 활성화해야 합니다. 자세한 내용은 스캔 유형 활성화를 참조하세요.

Lambda 함수 스캔을 활성화하면 Amazon Inspector는 계정에 `cloudtrail:CreateServiceLinkedChannel` 및 `cloudtrail:DeleteServiceLinkedChannel`이라는 다음과 같은 서비스 연결 채널을 생성합니다. Amazon Inspector는 이러한 채널을 관리하고 이를 사용하여 CloudTrail 이벤트의 스캔을 모니터링합니다. 이러한 채널을 사용하면 CloudTrail에 추적이 있는 것처럼 계정에서 CloudTrail 이벤트를 볼 수 있습니다. CloudTrail에서 자체 추적을 생성하여 계정의 이벤트를 관리하는 것이 좋습니다. 이러한 채널을 보는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 서비스 연결 채널 보기를 참조하세요.

> **Note**
>
> Amazon Inspector는 고객 관리형 키로 암호화된 Lambda 함수 스캔을 지원하지 않습니다. 이는 Lambda 표준 스캔 및 Lambda 코드 스캔에 적용됩니다.

Lambda 함수 스캔의 스캔 동작

Amazon Inspector는 활성화되면 계정에서 지난 90일 동안 간접적으로 호출되거나 업데이트된 모든 Lambda 함수를 스캔합니다. Amazon Inspector는 다음과 같은 경우에 Lambda 함수의 취약성 스캔을 시작합니다.

- Amazon Inspector에서 기존 Lambda 함수를 발견하는 즉시
- Lambda 서비스에 새 Lambda 함수를 배포하는 경우
- 기존 Lambda 함수 또는 해당 계층의 애플리케이션 코드 또는 종속성에 대한 업데이트를 배포하는 경우
- Amazon Inspector가 새로운 일반적인 취약성 및 노출(CVE) 항목을 데이터베이스에 추가하고, 해당 CVE가 함수와 관련이 있는 경우

Amazon Inspector는 Lambda 함수가 삭제되거나 검사에서 제외될 때까지 전체 기간 동안 각 Lambda 함수를 모니터링합니다.

Lambda 함수의 취약성을 마지막으로 확인한 시기는 계정 관리 페이지의 Lambda 함수 탭에서 또는 ListCoverage API를 사용하여 확인할 수 있습니다. Amazon Inspector는 다음 이벤트에 대한 응답으로 Lambda 함수의 마지막 스캔 시간 필드를 업데이트합니다.

- Amazon Inspector에서 Lambda 함수의 첫 번째 스캔을 완료한 경우
- Lambda 함수가 업데이트된 경우
- 함수에 영향을 미치는 새 CVE 항목이 Amazon Inspector 데이터베이스에 추가되어 Amazon Inspector에서 Lambda 함수를 다시 스캔하는 경우

지원되는 런타임 및 적합한 함수

Amazon Inspector는 Lambda 표준 스캔 및 Lambda 코드 스캔에 대해 다양한 런타임을 지원합니다. 스캔 유형별 지원되는 런타임 목록은 지원되는 런타임: Amazon Inspector Lambda 표준 스캔 및 지원되는 런타임: Amazon Inspector Lambda 코드 스캔 섹션을 참조하세요.

지원되는 런타임이 외에도 Lambda 함수가 Amazon Inspector 스캔 대상이 되려면 다음 기준을 충족해야 합니다.

- 함수가 지난 90일 이내에 간접적으로 호출되거나 업데이트되었습니다.
- 함수가 `$LATEST`로 표시됩니다.
- 함수가 태그 기준 스캔에서 제외되지 않았습니다.

> **Note**
>
> 지난 90일 이내에 간접적으로 호출되거나 수정되지 않은 Lambda 함수는 스캔에서 자동으로 제외됩니다. 자동으로 제외된 함수가 다시 간접적으로 호출되거나 Lambda 함수 코드가 변경되면 Amazon Inspector에서 해당 함수의 스캔을 재개합니다.

Amazon Inspector Lambda 표준 스캔

Amazon Inspector Lambda 표준 스캔은 Lambda 함수 코드 및 계층에 추가하는 애플리케이션 패키지 종속성의 소프트웨어 취약성을 식별합니다. 예를 들어 Lambda 함수에서 사용하는 `python-jwt` 패키지 버전에 알려진 취약성이 있을 경우 Lambda 표준 스캔은 해당 함수에 대한 결과를 생성합니다.

Amazon Inspector에서 Lambda 함수 애플리케이션 패키지 종속성의 취약성을 탐지한 경우 Amazon Inspector는 상세한 패키지 취약성 유형의 결과를 생성합니다.

스캔 유형 활성화에 대한 지침은 스캔 유형 활성화 섹션을 참조하세요.

> **Note**
>
> Lambda 표준 스캔은 Lambda 런타임 환경에 기본적으로 설치된 AWS SDK 종속성을 스캔하지 않습니다. Amazon Inspector는 함수 코드와 함께 업로드되거나 계층에서 상속된 종속성만 스캔합니다.

> **Note**
>
> Amazon Inspector Lambda 표준 스캔을 비활성화하면 Amazon Inspector Lambda 코드 스캔도 비활성화됩니다.

Lambda 표준 스캔에서 함수 제외

Lambda 함수에 태그를 추가하여 Amazon Inspector Lambda 표준 스캔에서 제외할 수 있습니다. 함수를 스캔에서 제외하면 조치할 수 없는 알림을 방지할 수 있습니다. 제외할 함수에 태그를 지정할 때 태그에는 다음과 같은 키-값 페어가 있어야 합니다.

- 키: `InspectorExclusion`
- 값: `LambdaStandardScanning`

이 주제에서는 스캔에서 제외할 함수에 태그를 지정하는 방법에 대해 설명합니다. Lambda에 태그를 추가하는 방법에 대한 자세한 내용은 Lambda 함수에서 태그 사용을 참조하세요.

스캔에서 함수를 제외하려면

1. 자격 증명을 사용하여 로그인한 다음 Lambda 콘솔(https://console.aws.amazon.com/lambda/)을 엽니다.
2. 탐색 창에서 함수를 선택합니다.
3. Amazon Inspector Lambda 표준 스캔에서 제외하려는 함수의 이름을 선택합니다.
4. 구성(Configuration)을 선택한 다음 태그(Tags)를 선택합니다.
5. 태그 관리, 새 태그 추가를 차례로 선택합니다.
   a. 키에 `InspectorExclusion`를 입력합니다.
   b. [값(Value)]에 `LambdaStandardScanning`을 입력합니다.
6. 저장을 선택합니다.

Amazon Inspector Lambda 코드 스캔

> **Important**
>
> 이 특성은 Lambda 함수의 스니펫을 캡처하여 탐지된 취약성을 강조 표시합니다. 이러한 스니펫에는 하드코딩된 자격 증명과 기타 민감한 자료가 표시될 수 있습니다.

이 특성을 사용하면 Amazon Inspector가 AWS 보안 모범 사례를 기반으로 Lambda 함수에서 애플리케이션 코드의 코드 취약성을 스캔하여 데이터 유출, 주입 결함, 암호화 누락 및 취약한 암호화를 탐지합니다. Amazon Inspector는 자동 추론 및 기계 학습을 사용하여 Lambda 함수 애플리케이션 코드를 평

가합니다. 또한 Amazon Q와 공동으로 개발한 내부 탐지기를 사용하여 정책 위반 및 취약성을 식별합니다.

Amazon Inspector는 Lambda 함수 애플리케이션 코드에서 취약성을 탐지하면 코드 취약성을 생성합니다. 이 조사 결과 유형에는 문제를 보여주는 코드 스니펫과 코드에서 문제를 찾을 수 있는 위치가 포함되어 있습니다. 또한 문제를 해결하는 방법도 제안합니다. 제안 사항에는 취약한 코드 줄을 대체하는 데 사용할 수 있는 플러그 앤 플레이 코드 블록이 포함되어 있습니다. 이러한 코드 수정은 해당 조사 결과 유형에 대한 일반 코드 수정 지침과 함께 제공됩니다.

코드 수정 제안은 자동 추론을 기반으로 합니다. 일부 코드 수정 제안은 의도한 대로 작동하지 않을 수 있습니다. 제안된 코드 수정을 사용할 경우 이에 따른 책임은 사용자에게 있습니다. 그러므로 제안된 코드 수정 방법을 사용하기 전에 항상 검토하세요. 코드가 의도한 대로 작동하는지 확인하기 위해 편집해야 할 수도 있습니다. 자세한 내용은 책임 있는 AI 정책을 참조하세요.

Lambda 코드 스캔을 활성화하려면 먼저 Lambda 표준 스캔을 활성화해야 합니다. 자세한 내용은 스캔 유형 활성화를 참조하세요. 이 특성을 지원하는 AWS 리전에 대한 자세한 내용은 리전별 특성 가용성 섹션을 참조하세요.

코드 취약성 조사 결과에서 코드 암호화

Amazon Q는 Lambda 코드 스캔을 사용하여 코드 취약성 조사 결과와 관련된 것으로 탐지된 코드 스니펫을 저장합니다. 기본적으로 Amazon Q는 코드를 암호화하는 데 사용되는 AWS 소유 키를 제어합니다. 하지만 Amazon Inspector API를 통해 암호화에 자체 고객 관리형 키를 사용할 수 있습니다. 자세한 내용은 조사 결과 코드에 대한 저장 중 암호화 섹션을 참조하세요.

Lambda 코드 스캔에서 함수 제외

Lambda 함수에 태그를 추가하여 Amazon Inspector Lambda 코드 스캔에서 제외할 수 있습니다. 함수를 스캔에서 제외하면 조치할 수 없는 알림을 방지할 수 있습니다. 제외할 함수에 태그를 지정할 때 태그에는 다음과 같은 키-값 페어가 있어야 합니다.

- 키 - `InspectorCodeExclusion`
- 값-`LambdaCodeScanning`

이 주제에서는 코드 스캔에서 제외할 함수에 태그를 지정하는 방법에 대해 설명합니다. Lambda에 태그를 추가하는 방법에 대한 자세한 내용은 Lambda 함수에서 태그 사용을 참조하세요.

코드 스캔에서 함수를 제외하려면

1. 자격 증명을 사용하여 로그인한 다음 Lambda 콘솔(https://console.aws.amazon.com/lambda/)을 엽니다.
2. 탐색 창에서 함수를 선택합니다.
3. Amazon Inspector Lambda 코드 스캔에서 제외할 함수의 이름을 선택합니다.
4. 구성(Configuration)을 선택한 다음 태그(Tags)를 선택합니다.
5. 태그 관리, 새 태그 추가를 차례로 선택합니다.
   a. 키에 `InspectorCodeExclusion`를 입력합니다.
   b. [값(Value)]에 `LambdaCodeScanning`을 입력합니다.
6. 저장을 선택합니다.

Amazon Inspector에서 스캔 유형 비활성화

스캔 유형을 비활성화하면 해당 스캔 유형으로 생성된 조사 결과에 액세스할 수 없게 됩니다. 스캔 유형을 다시 활성화하면 Amazon Inspector에서 모든 적격 리소스를 스캔하여 새로운 조사 결과를 생성합니다. 조사 결과를 보관하려면 조사 결과를 Amazon Simple Storage Service(Amazon S3) 버킷에 조사 결과 보고서로 내보낼 수 있습니다. 자세한 내용은 Amazon Inspector 조사 결과 보고서 내보내기 단원을 참조하십시오. 스캔 유형을 비활성화하면 스캔 유형을 비활성화한 AWS 계정에서 다음과 같은 변경 사항이 발생할 수 있습니다.

Amazon EC2 스캔

계정에 대해 Amazon Inspector Amazon EC2 스캔을 비활성화하면 Amazon 다음과 같은 SSM 연결이 삭제됩니다.

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

또한 Amazon Inspector SSM 플러그인도 모든 Windows 호스트에서 제거됩니다. 자세한 내용은 Windows EC2 인스턴스 스캔 단원을 참조하십시오.

Amazon ECR 스캔

계정에 대해 Amazon ECR 스캔을 비활성화하면 Amazon ECR 스캔 유형 계정이 Amazon Inspector를 사용하는 고급 스캔에서 Amazon ECR을 사용하는 기본 스캔으로 변경됩니다.

Lambda 표준 스캔

계정에 대해 Lambda 표준 스캔을 비활성화하면 Lambda 코드 스캔도 비활성화됩니다(해당 스캔 유형이 활성화된 경우). 또한 Lambda 표준 스캔을 활성화할 때 Amazon Inspector에서 생성한 CloudTrail 서비스 연결 채널도 삭제합니다.

Amazon Inspector 코드 보안

계정의 코드 보안을 비활성화하면 계정과 연결된 모든 통합, 프로젝트 및 스캔 구성이 삭제됩니다. 계정이 조직의 위임된 관리자인 경우 계정에 대한 코드 보안만 비활성화하고 멤버 계정은 독립 실행형 계정이 됩니다.

스캔 비활성화

계정에 대한 모든 스캔 유형을 비활성화하면 해당 AWS 리전의 해당 계정에 대한 Amazon Inspector가 비활성화됩니다. 자세한 내용은 Amazon Inspector 비활성화 단원을 참조하십시오.

다중 계정 환경에 대해 이 절차를 완료하려면 Amazon Inspector의 위임 관리자로 로그인한 상태에서 다음 단계를 수행하세요.

Console

스캔을 비활성화하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 페이지 오른쪽 상단의 AWS 리전 선택기를 사용하여 스캔을 비활성화할 리전을 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다.
4. 계정 탭을 선택하여 계정의 스캔 상태를 표시합니다.
5. 스캔을 비활성화할 각 계정의 확인란을 선택합니다.
6. 작업을 선택하고 비활성화 옵션에서 비활성화하려는 스캔 유형을 선택합니다.
7. (권장) 해당 스캔 유형을 비활성화 AWS 리전 하려는 각에서이 단계를 반복합니다.

API

Disable API 작업을 실행합니다. 요청에 스캔을 비활성화할 계정 ID를 제공하고, `resourceTypes`로 `EC2`, `ECR`, `LAMBDA`, `LAMBDA_CODE` 중 하나 이상을 제공하면 스캔이 비활성화됩니다.

Amazon EC2 인스턴스 운영 체제를 위한 Center for Internet Security(CIS) 스캔

Amazon Inspector CIS 스캔(CIS 스캔)은 Amazon EC2 인스턴스 운영 체제가 Center for Internet Security(CIS)에서 권장하는 모범 사례에 따라 구성되었는지 확인하기 위해 벤치마킹합니다. CIS 보안 벤치마크는 시스템을 안전하게 구성하기 위한 업계 표준 구성 기준 및 모범 사례를 제공합니다. 계정에 대해 Amazon Inspector EC2 스캔을 활성화한 후 CIS 스캔을 수행하거나 예약할 수 있습니다. Amazon EC2 스캔을 활성화하는 자세한 방법은 스캔 유형 활성화를 참조하세요.

> **Note**
>
> CIS 표준은 x86_64 운영 체제용입니다. 일부 검사는 평가되지 않거나 ARM 기반 리소스에 대한 잘못된 수정 지침을 반환할 수 있습니다.

Amazon Inspector는 인스턴스 태그와 정의된 스캔 일정에 따라 대상 Amazon EC2 인스턴스에 대해 CIS 스캔을 수행합니다. Amazon Inspector는 대상 인스턴스 각각에 대해 일련의 인스턴스 검사를 수행합니다. 검사할 때마다 시스템 구성이 특정 CIS 벤치마크 권장 사항을 충족하는지 평가합니다. 각 검사에는 CIS 검사 ID와 제목이 있으며, 이는 해당 플랫폼에 대한 CIS 벤치마크 권장 사항에 해당합니다. CIS 검사가 완료되면 결과를 확인하여 해당 시스템에 대해 어떤 인스턴스 검사가 통과, 건너뛰기 또는 실패했는지 확인할 수 있습니다.

> **Note**
>
> CIS 스캔을 수행하거나 예약하려면 보안 인터넷 연결이 필요합니다. 하지만 프라이빗 인스턴스에서 CIS 스캔을 실행하려면 VPC 엔드포인트를 사용해야 합니다.

주제

- Amazon Inspector CIS 스캔을 위한 Amazon EC2 인스턴스 요구 사항
- CIS 스캔 실행
- 를 사용하여 Amazon Inspector CIS 스캔을 관리하기 위한 고려 사항 AWS Organizations
- Amazon Inspector CIS 스캔에 사용되는 Amazon Inspector 소유 Amazon S3 버킷
- CIS 스캔 구성 생성

- CIS 스캔 결과 보기
- CIS 스캔 구성 편집
- CIS 스캔 결과 다운로드

Amazon Inspector CIS 스캔을 위한 Amazon EC2 인스턴스 요구 사항

Amazon EC2 인스턴스에서 CIS 스캔을 실행하려면 Amazon EC2 인스턴스가 다음 기준을 충족해야 합니다.

- 인스턴스 운영 체제가 CIS 스캔을 위해 지원되는 운영 체제 중 하나입니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.
- 인스턴스가 Amazon EC2 Systems Manager 인스턴스입니다. 자세한 내용은 AWS Systems Manager 사용 설명서에서 SSM Agent 작업을 참조하세요.
- Amazon Inspector SSM 플러그인이 인스턴스에 설치되어 있습니다. Amazon Inspector는 관리형 인스턴스에 이 플러그인을 자동으로 설치합니다.
- 인스턴스에는 SSM이 인스턴스를 관리할 수 있는 권한과 Amazon Inspector가 해당 인스턴스에 대해 CIS 스캔을 실행할 수 있는 권한을 부여하는 인스턴스 프로파일이 있습니다. 이러한 권한을 부여하려면 AmazonSSMManagedInstanceCore 및 AmazonInspector2ManagedCisPolicy 정책을 IAM 역할에 연결합니다. 그런 다음 IAM 역할을 인스턴스에 인스턴스 프로파일로 연결합니다. 인스턴스 프로파일 생성 및 연결에 대한 지침은 Amazon EC2 사용 설명서에서 IAM 역할 작업을 참조하세요.

> **Note**
>
> Amazon EC2 인스턴스에서 CIS 스캔을 실행하기 전에 Amazon Inspector 심층 검사를 활성화할 필요는 없습니다. Amazon Inspector 심층 검사를 비활성화하면 Amazon Inspector가 SSM 에이전트를 자동으로 설치하지만 더 이상 심층 검사를 실행하기 위해 SSM 에이전트가 간접적으로 호출되지 않습니다. 그러나 결과적으로 `InspectorLinuxDistributor-do-not-delete` 연결은 계정에 존재합니다.

프라이빗 Amazon EC2 인스턴스에서 CIS 스캔을 실행하기 위한 Amazon Virtual Private Cloud 엔드포인트 요구 사항

Amazon 네트워크를 통해 Amazon EC2 인스턴스에서 CIS 스캔을 실행할 수 있습니다. 하지만 프라이빗 Amazon EC2 인스턴스에서 CIS 스캔을 실행하려면 Amazon VPC 엔드포인트를 생성해야 합니다. Systems Manager용 Amazon VPC 엔드포인트를 생성할 때 다음 엔드포인트가 필요합니다.

- `com.amazonaws.region.ec2messages`
- `com.amazonaws.region.inspector2`
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ssm`
- `com.amazonaws.region.ssmmessages`

자세한 내용은 AWS Systems Manager 사용 설명서에서 Systems Manager를 위한 VPC 엔드포인트 생성을 참조하세요.

> **Note**
>
> 현재 일부 AWS 리전 는 `amazonaws.com.region.inspector2` 엔드포인트를 지원하지 않습니다.

CIS 스캔 실행

CIS 스캔은 온디맨드 방식으로 실행하거나 예약된 반복 스캔으로 실행할 수 있습니다. 스캔을 실행하려면 먼저 스캔 구성을 생성하세요.

스캔 구성을 생성할 때 대상 인스턴스에 사용할 태그 키-값 페어를 지정합니다. 조직의 Amazon Inspector 위임된 관리자인 경우 스캔 구성에 여러 개의 계정을 지정할 수 있으며, Amazon Inspector는 이러한 각 계정에서 지정된 태그가 있는 인스턴스를 찾습니다. 스캔에 대한 CIS 벤치마크 레벨을 선택하세요. 각 벤치마크에 대해 CIS는 다양한 환경에서 요구되는 다양한 보안 레벨을 위한 기준으로 설계된 레벨 1 및 레벨 2 프로파일을 지원합니다.

- 레벨 1 - 모든 시스템에서 구성할 수 있는 필수 기본 보안 설정을 권장합니다. 이 설정을 구현하면 서비스 중단이 거의 또는 전혀 발생하지 않습니다. 이러한 권장 사항의 목표는 시스템으로 유입되는 진입 지점의 수를 줄여 전반적인 사이버 보안 위험을 줄이는 것입니다.

- 레벨 2 - 보안이 중요한 환경을 위한 고급 보안 설정을 권장합니다. 이러한 설정을 구현하려면 비즈니스에 미칠 수 있는 위험을 최소화하기 위한 계획과 조정이 필요합니다. 이러한 권장 사항의 목표는 규정 준수를 달성하는 데 도움을 주는 것입니다.

레벨 2는 레벨 1을 확장한 것입니다. 레벨 2를 선택하면 Amazon Inspector에서 레벨 1 및 레벨 2에 권장되는 모든 구성을 확인합니다.

스캔에 대한 파라미터를 정의한 후에는 구성을 완료한 후 실행되는 일회성 스캔으로 실행할지, 아니면 반복 스캔으로 실행할지 여부를 선택할 수 있습니다. 반복 스캔은 원하는 시간에 매일, 매주 또는 매월 실행할 수 있습니다.

> **Tip**
>
> 스캔이 실행되는 동안 시스템에 영향을 미칠 가능성이 가장 낮은 날짜와 시간을 선택하는 것이 좋습니다.

를 사용하여 Amazon Inspector CIS 스캔을 관리하기 위한 고려 사항 AWS Organizations

조직에서 CIS 스캔을 실행하면 Amazon Inspector 위임된 관리자 멤버 계정은 서로 다른 방식으로 CIS 스캔 구성 및 스캔 결과와 상호 작용합니다.

Amazon Inspector 위임된 관리자가 CIS 스캔 구성 및 스캔 결과와 상호 작용하는 방식

위임된 관리자가 모든 계정 또는 특정 멤버 계정에 대해 스캔 구성을 생성하면 조직이 해당 구성을 소유하게 됩니다. 조직이 소유한 스캔 구성에는 조직 ID를 소유자로 지정하는 ARN이 있습니다.

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

위임된 관리자는 다른 계정에서 생성한 스캔 구성이라도 조직이 소유한 경우 관리할 수 있습니다.

위임된 관리자는 조직의 모든 계정에 대한 스캔 결과를 볼 수 있습니다.

위임된 관리자가 스캔 구성을 생성하고 SELF를 대상 계정으로 지정할 경우 위임된 관리자가 조직에서 탈퇴하더라도 스캔 구성을 계속 소유합니다. 그러나 위임된 관리자는 SELF를 대상으로 지정한 스캔 구성의 대상을 변경할 수는 없습니다.

> **Note**
>
> 위임된 관리자는 조직이 소유한 CIS 스캔 구성에 태그를 추가할 수 없습니다.

Amazon Inspector 멤버 계정이 CIS 스캔 구성 및 스캔 결과와 상호 작용하는 방식

멤버 계정이 CIS 스캔 구성을 생성하면 해당 구성을 소유하게 됩니다. 그러나 위임된 관리자가 해당 구성을 볼 수 있습니다. 멤버 계정이 조직에서 탈퇴하면 위임된 관리자는 더 이상 해당 구성을 볼 수 없습니다.

> **Note**
>
> 위임된 관리자는 멤버 계정이 생성한 스캔 구성을 편집할 수 없습니다.

멤버 계정, SELF를 대상으로 지정한 위임된 관리자 및 독립 실행형 계정은 모두 자신이 생성한 스캔 구성을 소유합니다. 이러한 스캔 구성에는 계정 ID를 소유자로 표시하는 ARN이 있습니다.

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

멤버 계정은 위임된 관리자가 예약한 CIS 스캔의 결과를 포함하여 자신의 계정에서 스캔 결과를 볼 수 있습니다.

Amazon Inspector CIS 스캔에 사용되는 Amazon Inspector 소유 Amazon S3 버킷

OVAL(Open Vulnerability and Assessment Language)은 컴퓨터 시스템의 머신 상태를 평가하고 보고하는 방법을 표준화하는 정보 보안 작업입니다. 다음 표에는 CIS 스캔에 사용되는 OVAL 정의가 포함된 Amazon Inspector 소유의 Amazon S3 버킷이 모두 나열되어 있습니다. Amazon Inspector는 CIS 스캔에 필요한 OVAL 정의 파일을 준비합니다. 필요한 경우 Amazon Inspector 소유 Amazon S3 버킷을 VPC에 허용 목록으로 지정해야 합니다.

> **Note**
>
> 다음 Amazon Inspector 소유의 Amazon S3 버킷에 대한 세부 정보는 변경되지 않습니다. 그러나 새로 지원되는 AWS 리전을 반영하기 위해 표가 업데이트될 수 있습니다. Amazon

Inspector 소유 Amazon S3 버킷은 다른 Amazon S3 작업이나 자체 Amazon S3 버킷에 사용할 수 없습니다.

CIS 버킷	AWS 리전
cis-datasets-prod-arn-5908f6f	유럽(스톡홀름)
cis-datasets-prod-bah-8f88801	Middle East (Bahrain)
cis-datasets-prod-bjs-0f40506	중국(베이징)
cis-datasets-prod-bom-435a167	아시아 태평양(뭄바이)
cis-datasets-prod-cdg-f3a9c58	유럽(파리)
cis-datasets-prod-cgk-09eb12f	아시아 태평양(자카르타)
cis-datasets-prod-cmh-63030b9	미국 동부(오하이오)
cis-datasets-prod-cpt-02c5c6f	아프리카(케이프타운)
cis-datasets-prod-dub-984936f	유럽(아일랜드)
cis-datasets-prod-fra-6eb96eb	유럽(프랑크푸르트)
cis-datasets-prod-gru-de69f99	남아메리카(상파울루)
cis-datasets-prod-hkg-8e30800	아시아 태평양(홍콩)
cis-datasets-prod-iad-8438411	미국 동부(버지니아 북부)
cis-datasets-prod-icn-f4eff1c	아시아 태평양(서울)
cis-datasets-prod-kix-5743b21	아시아 태평양(오사카)
cis-datasets-prod-lhr-8b1fbd0	유럽(런던)
cis-datasets-prod-mxp-7b1bbce	유럽(밀라노)
cis-datasets-prod-nrt-464f684	아시아 태평양(도쿄)

CIS 버킷	AWS 리전
cis-datasets-prod-osu-5bead6f	AWS GovCloud(미국 동부)
cis-datasets-prod-pdt-adadf9c	AWS GovCloud(미국 서부)
cis-datasets-prod-pdx-acfb052	미국 서부(오리건)
cis-datasets-prod-sfo-1515ba8	미국 서부(캘리포니아 북부)
cis-datasets-prod-sin-309725b	아시아 태평양(싱가포르)
cis-datasets-prod-syd-f349107	아시아 태평양(시드니)
cis-datasets-prod-yul-5e0c95e	캐나다(중부)
cis-datasets-prod-zhy-5a8eacb	중국(닝샤)
cis-datasets-prod-zrh-67e0e3d	유럽(취리히)

CIS 스캔 구성 생성

이 주제에서는 CIS 스캔 구성을 생성하는 방법에 대해 설명합니다.

CIS 스캔을 실행하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. AWS 리전 드롭다운을 사용하여 CIS 스캔을 실행할 AWS 리전을 선택합니다.
3. 탐색 창에서 온디맨드 스캔을 선택한 다음 CIS 스캔을 선택합니다.
4. 새 스캔 생성을 선택합니다.
5. 스캔 구성 이름에 스캔 구성 이름을 입력합니다.
6. 대상 리소스 태그에 스캔하려는 인스턴스의 키와 해당 값을 입력합니다. 각 키마다 최대 5개의 서로 다른 값을 지정하고 스캔에 포함할 태그는 총 25개까지 지정할 수 있습니다.
7. CIS 벤치마크 레벨에서 기본 보안 구성의 경우 레벨 1을 선택하고 고급 보안 구성의 경우 레벨 2를 선택할 수 있습니다.

8. 대상 계정에서 CIS 스캔에 포함할 계정을 지정합니다. 자세한 내용은 를 사용하여 Amazon Inspector CIS 스캔을 관리하기 위한 고려 사항 AWS Organizations 섹션을 참조하세요.

   계정이 위임된 관리자 계정인 경우 모든 계정 또는 계정 지정을 선택할 수 있습니다. 모든 계정 옵션은 조직의 모든 계정을 대상으로 합니다. 계정 지정은 조직의 개별 계정만 대상으로 합니다. 이 옵션을 선택하면 계정 번호를 쉼표로 구분하여 두 개 이상의 계정을 지정할 수 있습니다. 계정 ID 대신 SELF를 입력하여 계정에 대한 스캔 구성을 생성할 수도 있습니다.

   계정이 독립 실행형 계정이거나 조직의 멤버 계정인 경우 셀프를 선택하여 계정에 대한 스캔 구성을 생성할 수 있습니다.

9. 예약에서 스캔 구성 생성을 완료하는 즉시 실행되는 일회성 스캔 또는 지정한 시간에 실행되는 반복 스캔을 선택합니다.

10. 선택 사항을 검토한 다음 생성을 선택합니다.

CIS 스캔 결과 보기

Amazon Inspector는 실행되는 모든 스캔 구성에 대해 스캔 작업을 생성하고 고유한 스캔 ID로 스캔 결과를 수집합니다. CIS 스캔 결과는 90일 동안 제공됩니다. CIS 스캔 결과를 검사 또는 스캔한 리소스별로 볼 수 있습니다:

- 검사별로 집계된 스캔 결과 - 스캔 중에 수행된 개별 검사별로 스캔 결과를 그룹화합니다. 실패한 리소스, 건너뛴 리소스 또는 통과한 리소스 수에 대한 보고서가 각 검사별로 제공됩니다.
- 스캔한 리소스별로 집계된 스캔 결과 - 스캔하는 동안 스캔 대상이 되는 각 스캔 리소스별로 스캔 결과를 그룹화합니다. 각 리소스에 대해 실패, 건너뛰기 또는 통과한 검사 수에 대한 보고서가 제공됩니다.

이 주제에서는 CIS 스캔의 결과를 보는 방법에 대해 설명합니다.

스캔 결과를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. AWS 리전 드롭다운을 사용하여 CIS 스캔 구성을 생성한 AWS 리전을 선택합니다.
3. 탐색 창에서 온디맨드 스캔을 선택한 다음 CIS 스캔을 선택합니다.
4. 스캔 결과 탭을 선택합니다.

5. 예약자 열에서 보려는 스캔 예약 ID를 선택합니다. 또는 보려는 스캔 예약 ID가 있는 행을 선택한 다음 세부 정보 보기를 선택합니다.
6. 수행된 검사를 각각 보려면 검사를 선택하고, 스캔 중에 대상으로 지정된 스캔한 리소스를 각각 보려면 스캔된 리소스를 선택합니다.

예약된 CIS 스캔에 대한 세부 정보를 볼 수도 있습니다.

예약된 CIS 스캔에 대한 세부 정보를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. AWS 리전 드롭다운을 사용하여 CIS 스캔 구성을 생성한 AWS 리전을 선택합니다.
3. 탐색 창에서 온디맨드 스캔을 선택한 다음 CIS 스캔을 선택합니다.
4. 예약됨 탭을 선택합니다.
5. 스캔 구성 이름 열에서 보려는 스캔 구성의 이름을 선택합니다. 또는 보려는 스캔 구성이 있는 행을 선택한 다음 세부 정보 보기를 선택합니다.

CIS 스캔 구성 편집

이 주제에서는 CIS 스캔 구성을 편집하는 방법에 대해 설명합니다.

CIS 스캔 구성을 편집하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. AWS 리전 드롭다운을 사용하여 CIS 스캔 구성을 생성한 AWS 리전을 선택합니다.
3. 탐색 창에서 온디맨드 스캔을 선택한 다음 CIS 스캔을 선택합니다.
4. 예약됨 탭을 선택합니다.
5. 편집하려는 스캔 구성이 있는 행을 선택한 다음 편집을 선택합니다.

CIS 스캔 결과 다운로드

Amazon Inspector 콘솔 또는 API를 사용하여 CIS 스캔의 PDF 또는 CSV를 다운로드할 수 있습니다.

> **Note**
>
> 2024년 5월 3일 이후에 수집된 CIS 스캔에 대해서만 CIS 스캔 결과의 CSV 파일을 다운로드할 수 있습니다.

이 주제에서는 Amazon Inspector 콘솔을 사용하여 CIS 스캔을 다운로드하는 방법에 대해 설명합니다.

콘솔에서 CIS 스캔 결과를 다운로드하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. AWS 리전 드롭다운을 사용하여 CIS 스캔 구성을 생성한 AWS 리전을 선택합니다.
3. 탐색 창에서 온디맨드 스캔을 선택한 다음 CIS 스캔을 선택합니다.
4. 스캔 결과 탭을 선택합니다.
5. 예약자 열에서 보려는 스캔 예약 ID를 선택합니다. 또는 보려는 스캔 예약 ID가 있는 행을 선택한 다음 세부 정보 보기를 선택합니다.
6. 다운로드를 선택한 다음 PDF 또는 CSV를 선택합니다. 계정이 위임된 관리자 계정인 경우 계정 선택을 선택하여 특정 멤버 계정에 대한 결과를 다운로드할 수 있습니다.

Amazon Inspector 코드 보안

Amazon Inspector는 워크로드를 자동으로 검색하고 소프트웨어 취약성 및 의도하지 않은 네트워크 노출이 있는지 지속적으로 스캔하는 취약성 관리 서비스입니다. Amazon Inspector는 코드 보안을 사용하여 자사 애플리케이션 소스 코드, 타사 애플리케이션 종속성 및 코드형 인프라에서 취약성을 스캔합니다. Amazon Inspector 콘솔 또는 Amazon Inspector API를 통해 코드 보안을 활성화할 수 있습니다. 코드 보안을 활성화하면 코드 리포지토리에 스캔 구성을 생성하고 적용하여 스캔 빈도와 시기를 결정할 수 있습니다. 언제든지 스캔 구성을 보고, 편집하고, 삭제할 수 있습니다. 코드 보안을 사용할 수 있는 AWS 리전에 대한 자세한 내용은 리전 및 엔드포인트를 참조하세요. 요금에 대한 자세한 내용은 Amazon Inspector 요금을 참조하세요.

코드 보안을 위한 사전 조건

코드 보안 사용을 시작하려면 먼저 코드 보안을 활성화하고 데이터를 암호화하는 방법을 결정해야 합니다. 이는 통합 자격 증명, 코드 또는 통합, 코드 리포지토리 및 프로젝트와 관련된 기타 정보와 같은 정보일 수 있습니다. 기본적으로 데이터는 AWS 소유 키로 암호화됩니다. 즉, 서비스에서 키를 생성, 소유 및 관리합니다. 데이터를 암호화하는 데 사용되는 키를 소유하고 관리하려면 고객 관리형 KMS 키를 생성할 수 있습니다.

코드 보안 활성화

모든 자동 스캔 유형을 활성화하는 것과 동일한 방식으로 코드 보안을 활성화합니다. 자세한 내용은 스캔 유형 활성화를 참조하세요.

액세스할 고객 관리형 키 생성 AWS KMS

기본적으로 데이터는 AWS 소유 키로 암호화됩니다. 즉, 서비스에서 키를 생성, 소유 및 관리합니다. 데이터를 암호화하는 데 사용되는 키를 소유하고 관리하려면 고객 관리형 KMS 키를 생성할 수 있습니다. Amazon Inspector는 데이터와 상호 작용하지 않습니다. Amazon Inspector는 소스 코드 공급자의 리포지토리에서만 메타데이터를 수집합니다. 고객 관리형 KMS 키 생성 방법에 대한 자세한 내용은 AWS Key Management Service 사용 설명서의 KMS 키 생성을 참조하세요.

샘플 정책

고객 관리형 키를 생성할 때 다음 샘플 정책을 사용합니다.

> **Note**
>
> 다음 정책의 FAS 권한은 Amazon Inspector가 해당 API 직접 호출만 수행하도록 허용하므로 Amazon Inspector에만 적용됩니다.

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"
        }
      }
```

```
    }
  ]
}
```

KMS 키를 생성한 후 다음 Amazon Inspector API를 사용할 수 있습니다.

- UpdateEncryptionKey - 고객 관리형 KMS 키 사용을 구성하기 위한 스캔 유형으로 `resourceType` 및 `CODE`에 `CODE_REPOSITORY`를 사용합니다.
- GetEncryptionKey - KMS 키 구성의 검색을 구성하기 위한 스캔 유형으로 `resourceType` 및 `CODE`에 `CODE_REPOSITORY`를 사용합니다.
- ResetEncryptionKey - `resourceType` 및 `CODE` `CODE_REPOSITORY`에 대해를 사용하여 KMS 키 구성을 재설정하고 AWS 소유 KMS 키를 사용합니다.

Amazon Inspector 코드 리포지토리 간 통합 생성

이 섹션에는 Amazon Inspector와 코드 리포지토리 간의 통합을 생성하는 방법을 설명하는 주제가 포함되어 있습니다. 통합을 생성하면 코드 보안 페이지의 Amazon Inspector 콘솔에 모든 코드 리포지토리가 프로젝트로 나열됩니다. 이 섹션의 다른 주제에서는 통합 및 프로젝트에 액세스하는 방법을 설명합니다.

코드 보안은 최대 100,000개의 프로젝트만 가져오며 각 리포지토리의 기본 브랜치만 모니터링됩니다. 프로젝트는 최대 3개의 기본 스캔 구성과 연결할 수 있습니다.

코드 보안은 계정당 최대 100개의 통합만 지원합니다. 코드 보안 통합에는 위임된 관리자 계정/멤버 계정 관계에 대한 개념이 없습니다.

제한이 발생하지 않도록 통합에 동일한 호스트를 두 번 이상 사용하지 않는 것이 좋습니다.

GitHub SaaS, GitHub Enterprise Cloud 및 GitHub Enterprise Server와 통합하려면 퍼블릭 인터넷 액세스가 필요합니다.

> **Important**
>
> 서드 파티 통합은 보안 문제를 해결하는 등 어떤 이유로든 사전 통지 없이 일시적으로 또는 영구적으로 비활성화될 수 있습니다.

Amazon Inspector와 GitHub간의 통합 생성

이 주제에서는 Amazon Inspector와 GitHub간에 통합을 생성하는 방법을 설명합니다.

> **Note**
>
> 통합을 처음 생성하는 경우 2단계에서 기본 스캔 구성을 생성하라는 메시지가 표시됩니다. 스캔 구성을 생성할 때 스캔 빈도, 스캔 분석 및 스캔할 리포지토리를 선택합니다. 기본 스캔 구성을 생성하는 것은 일반 스캔 구성을 생성하는 것과 동일합니다. 그러나 기본 스캔 구성은 Amazon Inspector로 가져온 모든 신규 및 기존 프로젝트와 자동으로 연결됩니다. 기본 스캔 구성을 생성하려면 이 구성 계속을 선택합니다. 기본 스캔 구성은 한 번만 생성할 수 있습니다. 기본 스캔 구성을 생성하는 경우 기본 스캔 구성을 다시 생성하라는 메시지가 표시되지 않습니다. 기본 스캔 구성은 계정당 한 번만 생성할 수 있으며 조직당 한 번만 생성할 수 있습니다. 기본 스캔 구성을 구성하지 않으려면 구성 건너뛰기를 선택합니다. 그러나 다음에 통합을 생성할 때에 기본 스캔 구성을 생성하라는 메시지가 표시됩니다. 기본 스캔 구성을 생성하거나 기본 스캔 구성 생성을 건너뛰면 통합 세부 정보를 입력하는 통합 워크플로의 3단계로 이동합니다.

GitHub SaaS, GitHub Enterprise Cloud 및 GitHub Enterprise Server와 통합하려면 퍼블릭 인터넷 액세스가 필요합니다.

> **Note**
>
> Amazon Inspector는 기본 브랜치만 스캔하고 모니터링합니다. 새 기본 브랜치를 생성하면 Amazon Inspector가 새 기본 브랜치를 스캔하고 업데이트합니다.

> **Important**
>
> 통합 생성을 완료하기 전에 Amazon Inspector와 GitHub간의 연결을 승인하라는 메시지가 표시됩니다. 이 절차를 완료하려면 이 단계를 수행해야 합니다. 팝업을 닫으면 진행할 수 없습니다.

Amazon Inspector와 GitHub간의 통합을 생성하려면

1. 자격 증명을 사용하여 로그인합니다. Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.

2. 탐색 창에서 코드 보안을 선택합니다. 연결을 선택하고 GitHub를 선택합니다.
3. 통합 세부 정보에서 통합 이름을 입력하고 GitHub에 연결을 선택합니다.
4. 팝업에서 권한 부여를 선택하여 Amazon Inspector와 GitHub간에 연결을 생성합니다.
5. 성공 배너에서 GitHub 연결 생성 페이지로 이동을 선택합니다.
6. GitHub 애플리케이션의 설치 ID를 입력합니다. GitHub 애플리케이션을 설치한 경우 GitHub 앱 페이지 또는 GitHub 애플리케이션 URL 끝에서 GitHub의 설치 ID를 찾을 수 있습니다. GitHub 애플리케이션을 설치하지 않은 경우 새 앱 설치를 선택합니다. 그러면 GitHub 조직을 선택하고 리포지토리 범위를 지정하는 GitHub 위치로 이동합니다.
7. GitHub에 연결을 선택합니다.

통합을 생성한 후 Amazon Inspector가 액세스 토큰을 새로 고칠 수 없는 시나리오가 발생할 수 있습니다. 이는 통합 호스트를 사용할 수 없거나 Amazon Inspector에 다른 통신 문제가 발생하는 경우 발생할 수 있습니다. 문제를 해결하려면 코드 보안 페이지의 통합 탭에서 연결을 다시 인증할 수 있습니다. 상태 열에서 통합은 비활성으로 표시되고 Amazon Inspector는 재인증 옵션을 제공합니다. 재인증을 선택합니다. 연결 설정을 완료할 수 있는 통합 워크플로로 리디렉션됩니다.

통합에 대한 시스템 설정을 삭제하면 연결이 무기한 끊어질 수 있습니다. 이런 경우 통합을 삭제한 다음 새 통합을 만들어야 합니다. 통합을 삭제하면 통합과 연결된 모든 프로젝트 및 스캔 구성이 손실됩니다.

Amazon Inspector와 GitLab Self Managed간의 통합 생성

이 주제에서는 GitLab Self Managed에서 Amazon Inspector와 코드 리포지토리 간의 통합을 생성하는 방법을 설명합니다.

필수 정보

연결을 생성하는 경우 다음을 수행해야 합니다.

- 통합 이름 - 통합 본문에 추가된 이름입니다.
- 엔드포인트 URL - GitLab Self Managed 인스턴스에 액세스하는 데 사용되는 URL입니다.
- 개인 액세스 토큰 - 개인 액세스 토큰은 관리자 계정에서 GitLab Self Managed에 생성되며 `api`, `read_api`, `read_repository` 및 `write_repository` 범위를 포함해야 합니다.

> **Note**
>
> Amazon Inspector는 기본 브랜치만 스캔하고 모니터링합니다. 새 기본 브랜치를 생성하면 Amazon Inspector가 새 기본 브랜치를 스캔하고 업데이트합니다.

Amazon Inspector와 GitLab Self Managed간의 통합 생성

다음 절차에서는 GitLab Self Managed에서 Amazon Inspector와 코드 리포지토리 간에 연결을 생성하는 방법을 설명합니다.

> **Note**
>
> 통합을 처음 생성하는 경우 2단계에서 기본 스캔 구성을 생성하라는 메시지가 표시됩니다. 스캔 구성을 생성할 때 스캔 빈도, 스캔 분석 및 스캔할 리포지토리를 선택합니다. 기본 스캔 구성을 생성하는 것은 일반 스캔 구성을 생성하는 것과 동일합니다. 그러나 기본 스캔 구성은 Amazon Inspector로 가져온 모든 신규 및 기존 프로젝트와 자동으로 연결됩니다. 기본 스캔 구성을 생성하려면 이 구성 계속을 선택합니다. 기본 스캔 구성은 한 번만 생성할 수 있습니다. 기본 스캔 구성을 생성하는 경우 기본 스캔 구성을 다시 생성하라는 메시지가 표시되지 않습니다. 기본 스캔 구성은 계정당 한 번만 생성할 수 있으며 조직당 한 번만 생성할 수 있습니다. 기본 스캔 구성을 구성하지 않으려면 구성 건너뛰기를 선택합니다. 그러나 다음에 통합을 생성할 때 기본 스캔 구성을 생성하라는 메시지가 표시됩니다. 기본 스캔 구성을 생성하거나 기본 스캔 구성 생성을 건너뛰면 통합 세부 정보를 입력하는 통합 워크플로의 3단계로 이동합니다.

> **Important**
>
> 통합 생성을 완료하기 전에 Amazon Inspector와 GitLab 자체 관리형 간의 연결을 승인하라는 메시지가 표시됩니다. 이 절차를 완료하려면 이 단계를 수행해야 합니다. 팝업을 닫으면 진행할 수 없습니다.

GitLab 자체 관리형으로 연결을 생성하려면

1. 자격 증명을 사용하여 로그인합니다. Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다. 연결 대상을 선택한 다음 GitLab 자체 관리형을 선택합니다.

3. 통합 세부 정보에서 다음을 입력합니다.

   a. 통합 이름에 통합 본문에 추가된 이름을 입력합니다.
   b. 엔드포인트 URL에 GitLab 자체 관리형 인스턴스에 액세스하는 데 사용되는 URL을 입력합니다.
   c. 개인 액세스 토큰에 필요한 범위와 함께 개인 액세스 토큰을 입력합니다.
4. GitLab에 연결을 선택합니다.
5. 팝업 창에서 권한 부여를 선택하여 Amazon Inspector와 GitLab간의 연결 생성을 완료합니다.

통합을 생성한 후 Amazon Inspector가 액세스 토큰을 새로 고칠 수 없는 시나리오가 발생할 수 있습니다. 이는 통합 호스트를 사용할 수 없거나 Amazon Inspector에 다른 통신 문제가 발생하는 경우 발생할 수 있습니다. 문제를 해결하려면 코드 보안 페이지의 통합 탭에서 연결을 다시 인증할 수 있습니다. 상태 열에서 통합은 비활성으로 표시되고 Amazon Inspector는 재인증 옵션을 제공합니다. 재인증을 선택합니다. 연결 설정을 완료할 수 있는 통합 워크플로로 리디렉션됩니다.

통합에 대한 시스템 설정을 삭제하면 연결이 무기한 끊어질 수 있습니다. 이런 경우 통합을 삭제한 다음 새 통합을 만들어야 합니다. 통합을 삭제하면 통합과 연결된 모든 프로젝트 및 스캔 구성이 손실됩니다.

코드 리포지토리와의 통합 보기

이 주제에서는 Amazon Inspector 콘솔에서 통합을 보는 방법을 설명합니다.

Amazon Inspector 콘솔에서 통합을 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. [통합(Integrations)]을 선택합니다. 이 탭에서 구성된 모든 통합을 검토하고 모든 통합에 대한 기본 정보를 검토할 수 있습니다. 이 정보에는 통합 이름, 통합 상태 및 소스 코드 공급자 이름이 포함됩니다.

공급자에 대한 재인증

통합을 생성한 후 Amazon Inspector가 액세스 토큰을 새로 고칠 수 없는 시나리오가 발생할 수 있습니다. 이는 통합 호스트를 사용할 수 없거나 Amazon Inspector에 다른 통신 문제가 발생하는 경우 발생할 수 있습니다. 문제를 해결하려면 코드 보안 페이지의 통합 탭에서 연결을 다시 인증할 수 있습니다. 상

태 열에서 통합은 비활성으로 표시되고 Amazon Inspector는 재인증 옵션을 제공합니다. 재인증을 선택합니다. 연결 설정을 완료할 수 있는 통합 워크플로로 리디렉션됩니다.

통합에 대한 시스템 설정을 삭제하면 연결이 무기한 끊어질 수 있습니다. 이런 경우 통합을 삭제한 다음 새 통합을 만들어야 합니다. 통합을 삭제하면 통합과 연결된 모든 프로젝트 및 스캔 구성이 손실됩니다.

코드 리포지토리 보기

이 주제에서는 Amazon Inspector 콘솔에서 코드 리포지토리를 보는 방법을 설명합니다.

Amazon Inspector 콘솔에서 코드 리포지토리를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 코드 리포지토리를 선택합니다. 이 탭에서 프로젝트로 나열된 모든 코드 리포지토리를 검토하고 이에 대한 기본 정보를 검토할 수 있습니다. 이 정보에는 각 프로젝트의 이름과 스캔 상태가 포함됩니다. 프로젝트와 연결된 구성과 프로젝트가 마지막으로 스캔된 시기를 검토할 수도 있습니다. 검색 창에서 프로젝트를 필터링할 수도 있습니다.

프로젝트의 세부 정보 보기

이 주제에서는 Amazon Inspector 콘솔에서 프로젝트의 세부 정보를 보는 방법을 설명합니다. 계정이 조직의 위임된 관리자인 경우 멤버 계정에 속한 프로젝트의 세부 정보를 볼 수 있습니다.

Amazon Inspector 콘솔에서 코드 프로젝트를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 코드 리포지토리를 선택합니다. 이 탭에서 프로젝트로 나열된 모든 코드 리포지토리를 검토하고 이에 대한 기본 정보를 검토할 수 있습니다. 이 정보에는 각 프로젝트의 이름과 스캔 상태가 포함됩니다. 프로젝트와 연결된 구성과 프로젝트가 마지막으로 스캔된 시기를 검토할 수도 있습니다. 검색 창에서 프로젝트를 필터링할 수도 있습니다.
4. 프로젝트를 선택합니다. 또는 프로젝트를 선택하고 세부 정보 보기를 선택합니다. 프로젝트 세부 정보 화면에서 프로젝트에 대한 기본 정보를 검토할 수 있습니다. 이 정보에는 프로젝트의 이름과 ID, 통합 ARN이 포함됩니다. 여기에는 프로젝트를 스캔한 시기와 제공 유형에 대한 정보가 포함됩

니다. 프로젝트와 관련된 조사 결과를 검토하고 조사 결과를 내보내고 조사 결과에 대한 억제 규칙을 생성할 수도 있습니다.

통합 삭제

다음 절차에서는 Amazon Inspector 콘솔에서 통합을 삭제하는 방법을 설명합니다. 통합을 삭제하면 통합과 연결된 모든 프로젝트 및 스캔 구성이 손실됩니다.

Amazon Inspector 콘솔에서 통합을 삭제합니다.

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. [통합(Integrations)]을 선택합니다. 이 탭에서 구성된 모든 통합을 검토하고 모든 통합에 대한 기본 정보를 검토할 수 있습니다. 이 정보에는 통합 이름, 통합 상태 및 통합 공급자 유형이 포함됩니다.
4. 통합을 선택하고 삭제를 선택합니다.

스캔 구성 생성

스캔 구성을 생성하기 전에 Amazon Inspector와의 통합을 생성해야 합니다. 통합을 처음 생성할 때 기본 스캔 구성을 생성하라는 메시지가 표시됩니다. 이 주제에서는 일반 스캔 구성을 생성하는 방법에 대해 설명합니다. 기본 스캔 구성과 일반 스캔 구성의 차이점은 기본 스캔 구성이 새 프로젝트에 자동으로 연결된다는 것입니다. 기본 스캔 구성 생성을 건너뛸 수 있습니다.

코드 보안은 최대 500개의 일반 스캔 구성만 지원합니다. 코드 보안은 계정 및 조직당 1개의 기본 스캔 구성만 지원합니다. 스캔 구성만 최대 100,000개의 프로젝트와 연결할 수 있습니다.

프로젝트는 최대 총 4개의 스캔 구성과 연결할 수 있습니다. 여기에는 기본 스캔 구성이 생성된 경우 기본 스캔 구성이 포함됩니다. 조직의 스캔 구성에는 태그를 지정할 수 없습니다.

조직의 위임된 관리자가 스캔 구성을 생성하면 조직 수준에서 스캔 구성이 생성되고 조직의 모든 멤버 계정에 적용됩니다. 위임된 관리자가 기본 스캔 구성을 생성하는 경우에도 마찬가지입니다.

스캔 구성을 생성할 때 스캔 빈도, 스캔 분석 및 스캔할 리포지토리를 선택합니다. 스캔 빈도는 변경에 따라 주기적으로 또는 사용자 지정할 수 있습니다. 변경 기반 및 주기적 스캔을 사용하면 주기적 스캔을 활성화할 수 있습니다. 정기 스캔을 활성화하면 스캔 빈도를 스캔이 발생하는 요일 또는 월로 설정합니다. 사용자 지정 스캔을 사용하면 코드가 변경되고 주기적으로 스캔할 때 스캔을 활성화할 수 있습니다. 코드가 변경될 때 스캔을 활성화하는 경우 병합 및 풀 요청에 포함할 스캔 트리거를 지정합니다.

설정된 시간 내에 커밋 ID가 변경되지 않은 경우 스캔을 건너뛸 수 있습니다. 정기 스캔의 경우 1주일 동안 스캔 간에 커밋 ID가 변경되지 않은 경우 스캔을 건너뜁니다. 온디맨드 스캔의 경우 24시간 동안 스캔 간에 커밋 ID가 변경되지 않은 경우 스캔을 건너뜁니다.

> **Note**
>
> 스캔 구성에 병합 요청 및 풀 요청에 대한 트리거만 있는 경우 상위 25개의 중요하거나 높은 조사 결과만 소스 코드 관리 플랫폼에 표시됩니다. Amazon Inspector에는 아무것도 표시되지 않습니다.

일반 스캔 구성을 생성하려면

1. 자격 증명을 사용하여 로그인합니다. Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 구성 탭을 선택한 다음 스캔 구성 생성을 선택합니다.
4. 스캔 세부 정보에서 다음을 수행합니다.
   - 구성 이름에 스캔 구성 이름을 입력합니다.
5. 스캔 빈도에서 변경 기반 및 주기적 스캔 또는 사용자 지정 스캔 유형 및 트리거를 선택하여 코드를 스캔하는 빈도를 지정합니다.
   a. (옵션 1) 변경 기반 및 주기적 스캔을 선택하는 경우 주기적 스캔 활성화 또는 주기적 스캔 비활성화를 선택합니다.
      - 주기적 스캔 활성화를 선택한 경우 코드를 스캔할 주와 일을 선택하여 스캔 빈도를 설정합니다.
   b. (옵션 2) 사용자 지정 스캔을 선택한 경우 코드가 변경될 때 스캔을 활성화할지 여부와 주기적 스캔을 선택합니다.
      i. 코드가 변경되면 스캔 활성화를 선택하고 코드가 변경되면 스캔 비활성화를 선택합니다. 코드가 변경될 때 스캔 활성화를 선택하는 경우 드롭다운에서 스캔이 트리거되는 시기를 지정합니다.
      ii. 주기적 스캔 활성화 또는 주기적 스캔 비활성화를 선택합니다. 정기 스캔 활성화를 선택한 경우 코드를 스캔할 주와 일을 선택하여 스캔 빈도를 설정합니다. 이벤트 기반 트리거를 스캔할 수도 있습니다. 이러한 이벤트에는 새 풀 요청이 기본 브랜치에 대해 처음 열리고 커밋이 병합되거나 기본 브랜치로 푸시되는 경우가 포함됩니다. 기존 풀 요청에 대

한 후속 업데이트 또는 개정에서는 스캔이 트리거되지 않습니다. 새 스캔을 트리거하려면 풀 요청을 닫았다가 다시 엽니다.

6. 스캔 분석에서 전체 스캔 분석 또는 사용자 지정 스캔 분석을 구성할지 여부를 결정합니다.

   a. (옵션 1) 스캔 분석 완료를 선택하면 다음 스캔 분석을 모두 적용합니다.

      - 정적 애플리케이션 보안 테스트 - 소스 코드의 취약성을 분석합니다.
      - IaC 스캔 - 인프라를 구성하고 프로비저닝하는 스크립트와 코드를 분석합니다.
      - 정적 소프트웨어 구성 분석 - 애플리케이션의 오픈 소스 패키지를 검사합니다.

   b. (옵션 2) 사용자 지정 스캔 분석을 선택하는 경우 드롭다운 메뉴에서 이전에 언급한 스캔 분석 유형을 하나 이상 선택해야 합니다.

7. (선택 사항) 태그에서 프로젝트에 적용할 키-값 페어를 생성합니다. 최대 50개의 태그를 생성할 수 있습니다.
8. 다음을 선택합니다.
9. 리포지토리 선택에서 모든 리포지토리 또는 특정 리포지토리를 선택합니다.

   a. (옵션 1) 모든 리포지토리를 선택하면 기존 리포지토리에 대한 스캔이 활성화됩니다.

   b. (옵션 2) 특정 리포지토리를 선택하면 지정한 리포지토리에 대해서만 스캔이 활성화됩니다.

10. 다음을 선택합니다.
11. 선택 사항을 살펴본 후 스캔 구성 만들기를 선택합니다.

> **Note**
>
> 일반 스캔 구성은 기존의 모든 코드 리포지토리에만 적용됩니다. 새 코드 리포지토리에는 적용되지 않습니다.

스캔 구성 보기

다음 절차에서는 Amazon Inspector 콘솔에서 스캔 구성을 보는 방법을 설명합니다.

> **Note**
>
> 조직 수준에서 스캔 구성을 볼 때 코드 보안 화면의 일부 세부 정보는 AWS 계정을 반영하도록 달라집니다.

스캔 구성에 대한 세부 정보를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 구성을 선택하여 스캔 구성 목록을 봅니다. 위임된 관리자인 경우 목록에 조직의 스캔 구성이 포함됩니다. 각 스캔 구성의 이름과 각 스캔 구성을 생성한 사람(AWS 계정 ID 또는 조직 ID)을 볼 수 있습니다. 구성에 적용되는 스캔 유형과 스캔 분석 유형을 볼 수도 있습니다. 검색 창의 여러 필드를 기준으로 스캔 구성을 필터링할 수도 있습니다.

스캔 구성에 대한 세부 정보 보기

다음 절차에서는 Amazon Inspector 콘솔에서 스캔 구성에 대한 세부 정보를 보는 방법을 설명합니다.

스캔 구성에 대한 세부 정보를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 구성을 선택합니다.
4. 세부 정보를 볼 구성을 선택합니다. 스캔 구성 세부 정보 화면은 스캔 구성의 개요를 제공합니다. 이 화면에서 스캔 구성 ARN, 활성화된 스캔 빈도 유형, 활성화된 스캔 분석 유형을 볼 수 있습니다. 이 화면에서 스캔 구성을 삭제할 수도 있습니다. 조직에 속한 스캔 구성을 보는 경우 이 화면에서도 편집할 수 있습니다.

스캔 구성 편집

스캔 구성은 언제든지 편집할 수 있습니다. 스캔 구성을 편집할 때 스캔 빈도, 스캔 분석, 태그 및 스캔할 리포지토리를 변경할 수 있습니다. 예를 들어 스캔 구성을 편집하여 특정 리포지토리에 대한 스캔을 일시 중지합니다. 다음 절차에서는 스캔 구성을 편집하는 방법을 설명합니다.

CIS 스캔 구성을 편집하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 구성을 선택합니다.

4. 편집할 구성을 선택하고 편집을 선택합니다. 편집할 구성을 선택한 다음 편집을 선택할 수 있습니다.

스캔 구성 삭제

스캔 구성은 언제든지 삭제할 수 있습니다. 이 주제에서는 CIS 스캔 구성을 삭제하는 방법에 대해 설명합니다.

스캔 구성을 삭제하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 구성을 선택합니다.
4. 삭제하려는 구성을 선택한 다음, 삭제를 선택합니다. 삭제하려는 구성을 선택한 다음, 삭제를 선택합니다.

온디맨드 스캔 수행

프로젝트에 대해 온디맨드를 수행할 수 있습니다. 온디맨드 스캔을 수행하면 구성된 모든 스캔 구성의 조합이 선택한 프로젝트에 적용됩니다. 계정이 조직의 위임된 관리자 계정인 경우 멤버 계정에 속한 프로젝트에 대해 온디맨드 스캔을 수행할 수 있습니다. 다음 절차에서는 Amazon Inspector 콘솔에서 온디맨드 스캔을 수행하는 방법을 설명합니다.

온디맨드 스캔을 수행하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 코드 보안을 선택합니다.
3. 코드 리포지토리를 선택합니다.
4. 스캔할 프로젝트를 선택한 다음 온디맨드 스캔을 선택합니다.

Amazon Inspector 코드 보안에 지원되는 언어

이 주제에는 Amazon Inspector 코드 보안에 지원되는 언어가 포함되어 있습니다.

SAST에 지원되는 언어

- C#(.Net 6.0 이상을 제외한 모든 버전이 권장됨)
- C(C11 또는 이전)
- C++(C++ 17 또는 이전)
- Go(Go 1.18만 해당)
- Java(Java 17 또는 이전)
- JavaScript(EMCMAScript 2021 이전)
- JSX(React 17 이전)
- Kotlin(Kotlin 2.0 또는 이전)
- PHP(PHP 8.2 또는 이전)
- Python(Python 3 시리즈 내에서 Python 3.11 또는 이전)
- Ruby(Ruby 2.7 및 3.2만 해당)
- Rust
- Scala(Scala 3.2.2 또는 이전)
- Shell
- TSX
- TypeScript(모든 버전)

소프트웨어 구성 분석에 지원되는 언어

- Go(Go 1.18만 해당)
- Java(Java 17 또는 이전)
- JavaScript(EMCMAScript 2021 이전)
- PHP(PHP 8.2 또는 이전)
- Python(Python 3 시리즈 내에서 Python 3.11 또는 이전)
- .Net
- Ruby(Ruby 2.7 및 3.2만 해당)
- Rust

코드형 인프라의 언어

- AWS CDK (Python 및 TypeScript)
- CloudFormation(2010–09–09)
- Terraform(1.6.2 또는 이전)

코드 보안 비활성화

코드 보안 비활성화에 대한 자세한 내용은 스캔 유형 비활성화를 참조하세요.

Amazon Inspector 조사 결과 이해

Amazon Inspector는 Amazon EC2 인스턴스, Amazon ECR의 컨테이너 이미지 및 Lambda 함수에서 수정이 가능하거나 수정이 보류 중인 취약성을 탐지하면 조사 결과를 생성합니다. 또한 자사 애플리케이션 소스 코드, 타사 애플리케이션 종속성 및 코드형 인프라에서 탐지된 코드 취약성에 대한 조사 결과를 생성합니다. 조사 결과는 AWS 리소스 중 하나에 영향을 미치는 취약성에 대한 자세한 보고서입니다.

결과는 취약성의 이름을 따서 명명되며 심각도 등급, 영향을 받는 AWS 리소스 및 비 AWS 리소스에 대한 정보, 탐지된 취약성을 해결하는 방법을 설명하는 세부 정보를 제공합니다. Amazon Inspector는 사용자가 수정할 때까지 모든 활성 조사 결과를 저장합니다.

리소스가 삭제되거나 종료되거나 더 이상 스캔할 수 없는 경우 Amazon Inspector는 리소스와 연결된 조사 결과를 자동으로 닫은 다음 3일 후에 조사 결과를 삭제합니다. 다른 이유로 조사 결과가 종료된 경우 30일 후에 삭제됩니다.

> **Note**
>
> Amazon Inspector는 취약성을 유발한 문제가 다시 발생할 경우 조사 결과 종결 후 7일 이내에 해결된 조사 결과를 다시 열 수 있습니다.

Amazon Inspector를 비활성화하면 24시간 후에 조사 결과가 제거됩니다. 리소스가 종료되면 리소스와 관련된 모든 조사 결과는 3일 후에 제거됩니다. 스캔이 더 이상 적합하지 않은 리소스에 연결된 모든 조사 결과에 대해서도 마찬가지입니다. 가 계정을 AWS 일시 중지하면 90일 후에 조사 결과가 제거됩니다. 중지된 인스턴스에 대한 조사 결과는 활성 상태로 유지됩니다.

조사 결과 상태

Amazon Inspector는 조사 결과를 다음 상태로 분류합니다.

활성

Amazon Inspector는 해결되지 않은 조사 결과를 활성으로 분류합니다.

억제됨

Amazon Inspector는 하나 이상의 억제 규칙이 적용된 조사 결과를 억제됨으로 분류합니다.

종결됨

조사 결과가 해결되면 Amazon Inspector는 조사 결과를 종결됨으로 분류합니다.

주제

- Amazon Inspector 조사 결과 유형
- Amazon Inspector 조사 결과 보기
- Amazon Inspector 조사 결과에 대한 세부 정보 보기
- Amazon Inspector 점수 보기 및 취약성 인텔리전스 세부 정보 이해
- Amazon Inspector 조사 결과의 심각도 수준 이해

Amazon Inspector 조사 결과 유형

이 단원에서는 Amazon Inspector의 다양한 조사 결과 유형에 대해 설명합니다.

주제

- 패키지 취약성
- 코드 취약성
- 네트워크 연결성

패키지 취약성

패키지 취약성 조사 결과는 사용자 AWS 환경에서 일반적인 취약성 및 노출(CVE)에 표시된 소프트웨어 패키지를 식별합니다. 공격자는 이러한 패치되지 않은 취약성을 악용하여 데이터의 기밀성, 무결성 또는 가용성을 손상시키거나 다른 시스템에 액세스할 수 있습니다. CVE 시스템은 공개적으로 알려진 정보 보안 취약성 및 노출도에 대한 참조 방법입니다. 자세한 내용은 https://www.cve.org/를 참조하세요.

Amazon Inspector는 EC2 인스턴스, ECR 컨테이너 이미지 및 Lambda 함수에 대한 패키지 취약성 탐지 조사 결과를 생성할 수 있습니다. 패키지 취약성 조사 결과에는 이 조사 결과 유형에만 적용되는 추가 세부 정보가 있는데, 바로 Inspector 점수 및 취약성 인텔리전스입니다.

코드 취약성

코드 취약성 조사 결과는 악용될 수 있는 코드 라인을 식별하는 데 도움이 됩니다. 코드 취약성에는 암호화 누락, 데이터 유출, 주입 결함, 취약한 암호화 등이 있습니다. Amazon Inspector는 Lambda 함수 스캔 및 코드 보안 특성을 통해 코드 취약성 조사 결과를 생성합니다.

Amazon Inspector는 애플리케이션 코드의 전반적인 보안 규정 준수를 분석하는 자동 추론 및 기계 학습을 사용하여 Lambda 함수 애플리케이션 코드를 평가합니다. Amazon Q와 협력하여 개발된 내

부 탐지기를 기반으로 정책 위반 및 취약성을 식별합니다. 가능한 탐지 목록은 Amazon Q Detector Library를 참조하세요.

코드 스캔은 코드 스니펫을 캡처하여 탐지된 취약성을 강조 표시합니다. 예를 들어, 코드 스니펫에는 하드코딩된 보안 인증 또는 기타 민감한 자료가 일반 텍스트로 표시될 수 있습니다. Amazon Q는 코드 취약성과 연결된 코드 조각을 저장합니다. 기본적으로 코드는 AWS 소유 키로 암호화됩니다. 그러나 이 정보를 더 잘 제어하려면 고객 관리형 키를 생성하여 코드를 암호화할 수 있습니다. 자세한 내용은 조사 결과 코드에 대한 저장 중 암호화 섹션을 참조하세요.

> **Note**
>
> 조직의 위임된 관리자는 멤버 계정에 속한 코드 조각을 볼 수 없습니다.

네트워크 연결성

네트워크 연결성 조사 결과는 사용자 환경에 Amazon EC2 인스턴스에 대한 오픈 네트워크 경로가 있음을 나타냅니다. 이러한 조사 결과는 인터넷 게이트웨이(Application Load Balancer 또는 Classic Load Balancer 뒤에 있는 인스턴스 포함), VPC 피어링 연결 또는 가상 게이트웨이를 통한 VPN 등의 VPC 엣지에서 TCP 및 UDP 포트에 연결할 수 있는 경우에 나타납니다. 이러한 조사 결과는 잘못 관리된 보안 그룹, 액세스 제어 목록 또는 인터넷 게이트웨이와 같이 지나치게 허용적인 네트워크 구성이나 잠재적으로 악의적인 액세스를 허용할 수 있는 네트워크 구성을 강조합니다.

Amazon Inspector는 Amazon EC2 인스턴스에 대한 네트워크 연결성 조사 결과만 생성합니다. Amazon Inspector가 활성화되면 12시간마다 네트워크 연결성 조사 결과를 스캔합니다.

Amazon Inspector는 네트워크 경로를 스캔할 때 다음 구성을 평가합니다.

- Amazon EC2 인스턴스:
- Application Load Balancers.
- Direct Connect
- Elastic Load Balancer
- 탄력적 네트워크 인터페이스
- 인터넷 게이트웨이
- 네트워크 액세스 제어 목록
- 라우팅 테이블

- 보안 그룹
- 서브넷,
- 가상 프라이빗 클라우드
- 가상 프라이빗 게이트웨이
- VPC 엔드포인트
- 게이트웨이 VPC 엔드포인트
- VPC 피어링 연결
- VPN 연결

Amazon Inspector 조사 결과 보기

조사 결과는 Amazon Inspector 콘솔과 Amazon Inspector `ListFindings` API를 통해 확인할 수 있습니다. Amazon Inspector 콘솔의 대시보드와 조사 결과 화면에서 모든 조사 결과를 볼 수 있습니다. 기본적으로 이러한 화면에는 활성 및 중요 조사 결과만 표시됩니다. 그러나 조사 결과를 필터링하거나 범주별로 조사 결과를 보도록 선택할 수 있습니다. 이러한 통합을 활성화하면 Security Hub CSPM 및 Amazon ECR에서 일부 결과를 볼 수도 있습니다. 이 섹션의 절차에서는 Amazon Inspector 콘솔과 Amazon Inspector `ListFindings` API에서 조사 결과를 보는 방법에 대해 설명합니다.

Console

Amazon Inspector 조사 결과를 보려면

1. 자격 증명을 사용하여 로그인합니다. Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. (선택 사항) 탐색 창에서 대시보드를 선택합니다. 대시보드에는 환경 적용 범위에 대한 개요와 활성 및 중요 조사 결과만 표시됩니다.
3. (선택 사항) 탐색 창에서 조사 결과를 선택합니다. 이 화면에는 모든 활성 조사 결과가 나열됩니다. 필터 기준을 사용하여 특정 조사 결과를 볼 수 있습니다. 목록에서 조사 결과를 제외하려면 억제 규칙을 생성합니다. 조사 결과에 대한 세부 정보를 보려면 조사 결과 이름을 선택합니다.
4. (선택 사항) 탐색 창에서 다음 옵션 중 하나를 선택하여 조사 결과를 범주별로 확인합니다.
   - 취약성별 - 가장 중요한 조사 결과와 함께 취약성이 표시됩니다.
   - 계정별 - 가장 중요한 조사 결과와 함께 계정이 표시됩니다. 이 범주는 위임된 관리자만 사용할 수 있습니다.

- 인스턴스별 - 가장 중요한 조사 결과와 함께 Amazon EC2 인스턴스가 표시됩니다. 이 범주에는 네트워크 가용성에 대한 정보가 포함되지 않습니다.
- 컨테이너 이미지별 - 가장 중요한 조사 결과와 함께 Amazon ECR 컨테이너 이미지가 표시됩니다. 이 범주는 컨테이너 이미지에 대한 기본 정보도 제공합니다. 여기에는 배포된 Amazon ECS 작업 수 및 Amazon EKS 포드 수와 같은 세부 정보도 포함됩니다. 이 화면에서는 지난 24시간 동안 실행되고 중지된 작업/포드 수를 확인할 수 있습니다.
- 컨테이너 리포지토리별 - 가장 중요한 조사 결과와 함께 컨테이너 리포지토리가 표시됩니다.
- Lambda 함수별 - 가장 중요한 조사 결과와 함께 Lambda 함수가 표시됩니다.

API

Amazon Inspector 조사 결과를 보려면

- ListFindings API 작업을 실행합니다. 요청에서 특정 조사 결과를 반환하도록 filterCriteria를 지정할 수 있습니다.

Amazon Inspector 조사 결과에 대한 세부 정보 보기

이 섹션의 절차에서는 Amazon Inspector 조사 결과에 대한 세부 정보를 보는 방법에 대해 설명합니다.

조사 결과 세부 정보를 확인하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 조사 결과를 확인할 리전을 선택합니다.
3. 탐색 창에서 조사 결과를 선택하여 조사 결과 목록을 표시합니다.
4. (선택 사항) 필터 막대를 사용하여 특정 조사 결과를 선택합니다. 자세한 내용은 Amazon Inspector 조사 결과 필터링 단원을 참조하십시오.
5. 조사 결과를 선택하여 세부 정보 패널을 확인합니다.

조사 결과 세부 정보 패널에는 조사 결과의 기본 식별 특성이 포함되어 있습니다. 이 기능은 조사 결과 제목과 식별된 취약성에 대한 기본 설명, 제안된 해결 방법, 심각도 점수입니다. 점수에 대한 자세한 내용은 Amazon Inspector 조사 결과의 심각도 수준 이해 섹션을 참조하세요.

조사 결과에 제공되는 세부 정보는 조사 결과 유형 및 영향을 받는 리소스에 따라 다릅니다.

모든 결과에는 조사 결과가 식별된 AWS 계정 ID 번호, 심각도, 조사 결과 유형, 조사 결과가 생성된 날짜, 해당 리소스에 대한 세부 정보가 포함된 영향을 받는 리소스 섹션이 포함됩니다.

조사 결과 유형에 따라 조사 결과에 제공되는 해결 방법 및 취약성 인텔리전스 정보가 결정됩니다. 조사 결과 유형에 따라 다른 조사 결과 세부 정보가 제공됩니다.

패키지 취약성

패키지 취약성 조사 결과는 EC2 인스턴스, ECR 컨테이너 이미지 및 Lambda 함수에 제공됩니다. 자세한 내용은 패키지 취약성 섹션을 참조하세요.

패키지 취약성 조사 결과에는 Amazon Inspector 점수 보기 및 취약성 인텔리전스 세부 정보 이해도 포함됩니다.

이 결과 유형에 대한 세부 정보는 다음과 같습니다.

- 수정 버전 있음 – 영향을 받는 패키지의 최신 버전에서 취약성이 수정되었는지 여부를 나타냅니다. 다음 값 중 하나를 사용합니다.
  - YES - 영향을 받는 모든 패키지에 수정 버전이 있습니다.
  - NO - 영향을 받는 패키지에 수정 버전이 없습니다.
  - PARTIAL - 영향을 받는 패키지 중 하나 이상(전부는 아님)에 수정 버전이 있습니다.
- 공격 가능 - 취약성에 알려진 악용 사례가 있음을 나타냅니다.
  - YES - 환경에서 발견된 취약성에 알려진 악용 사례가 있습니다. Amazon Inspector에서는 환경 내에서 발생하는 악용 사례를 파악할 수 없습니다.
  - NO - 해당 취약성에 알려진 악용 사례가 없습니다.
- 영향을 받는 패키지 - 조사 결과에서 취약한 것으로 식별된 각 패키지와 각 패키지의 세부 정보가 나열됩니다.
- Filepath - 조사 결과와 관련된 EBS 볼륨 ID 및 파티션 번호입니다. 이 필드는 에이전트 없는 스캔을 사용하여 스캔한 EC2 인스턴스의 조사 결과에 있습니다.
- 설치된 버전/수정된 버전 - 현재 설치된 패키지 중에서 취약성이 탐지된 버전 번호입니다. 설치된 버전 번호를 슬래시(/) 뒤의 값과 비교하세요. 두 번째 값은 탐지된 취약성을 수정한 패키지의 버전 번호로, 조사 결과와 관련된 일반적인 취약성 및 노출(CVE) 또는 권고에 따라 제공됩니다. 여러 버전에서 취약성이 수정된 경우 이 필드에는 수정 사항이 포함된 최신 버전이 나열됩니다. 수정 사항이 없는 경우 이 값은 None available입니다.

> **Note**
>
> Amazon Inspector에서 이 필드를 조사 결과에 포함시키기 전에 조사 결과가 발견된 경우 이 필드의 값은 비어 있습니다. 하지만 수정 사항이 있을 수 있습니다.

- 패키지 관리자 - 이 패키지를 구성하는 데 사용되는 패키지 관리자입니다.
- 해결 - 업데이트된 패키지 또는 프로그래밍 라이브러리를 통해 수정 사항이 제공되는 경우 이 섹션에는 업데이트를 위해 실행할 수 있는 명령이 포함됩니다. 제공된 명령을 복사하여 사용 환경에서 실행할 수 있습니다.

> **Note**
>
> 해결 명령은 공급업체 데이터 피드에서 제공되며 시스템 구성에 따라 달라질 수 있습니다. 자세한 지침은 조사 결과 참조 또는 운영 체제 설명서를 참조하세요.

- 취약성 세부 정보 - 조사 결과에서 식별된 CVE와 관련하여 NVD(National Vulnerability Database), REDHAT 또는 다른 OS 공급업체 등 Amazon Inspector 선호 소스로 연결되는 링크를 제공합니다. 또한 조사 결과에 대한 심각도 점수도 확인할 수 있습니다. 심각도 점수에 대한 자세한 내용은 Amazon Inspector 조사 결과의 심각도 수준 이해 섹션을 참조하세요. 점수 벡터를 포함하여 다음과 같은 점수가 포함됩니다.
  - 공격 예측 점수 시스템(EPSS) 점수
  - Inspector 점수
  - Amazon CVE의 CVSS 3.1
  - NVD의 CVSS 3.1
  - NVD의 CVSS 2.0(해당하는 경우, 이전 CVE의 경우)
- 관련 취약성 - 조사 결과와 관련된 다른 취약성을 지정합니다. 일반적으로 이들은 동일한 패키지 버전에 영향을 미치는 다른 CVE이거나 공급업체에서 결정한 조사 결과 CVE와 동일한 그룹 내의 다른 CVE입니다.
- 영향을 받는 리소스 - 레지스트리, 리포지토리, 리소스 유형, 이미지 ID 및 이미지 운영 체제에 대한 정보를 포함합니다. 또한 이미지가 마지막으로 푸시된 시간, 배포된 Amazon ECS 작업 및 Amazon EKS 포드 수, 지난 24시간 동안 이미지가 마지막으로 사용된 시간 등의 정보도 포함됩니다. Amazon ECS 작업 및 Amazon EKS 포드를 배포한 경우 필드 값을 선택하여 세부 정보를 볼 수 있습니다. 그러면 클러스터 ARN, 지난 24시간 동안 리소스가 마지막으로 사용된 시간, 리소스의 실행 및 중지 수, 워크로드 이름 및 유형과 같은 정보를 볼 수 있는 화면으로 이동합니다.

코드 취약성

코드 취약성 조사 결과는 Lambda 함수에만 사용할 수 있습니다. 자세한 내용은 코드 취약성 섹션을 참조하세요. 이 결과 유형에 대한 세부 정보는 다음과 같습니다.

- 수정 버전 있음 - 코드 취약성의 경우 이 값은 항상 YES입니다.
- 탐지기 이름 - 코드 취약성을 탐지하는 데 사용되는 Amazon Q 탐지기의 이름입니다. 가능한 탐지 목록은 Q 탐지기 라이브러리를 참조하세요.
- 탐지기 태그 - 탐지기와 관련된 Amazon Q 태그입니다. Amazon Q는 태그를 사용하여 탐지를 분류합니다.
- 관련 CWE - 코드 취약성과 관련된 CWE(Common Weakness Enumeration) ID입니다.
- 파일 경로 - 코드 취약성이 있는 파일 위치입니다.
- 취약성 위치 - Lambda 코드 스캔 코드 취약성의 경우 이 필드에는 Amazon Inspector에서 취약성을 발견한 정확한 코드 줄이 표시됩니다.
- 제안된 해결 방법 - 조사 결과를 해결하기 위한 코드 편집 방법을 제안합니다.

네트워크 연결성

네트워크 연결성 조사 결과는 EC2 인스턴스에만 사용할 수 있습니다. 자세한 내용은 네트워크 연결성 섹션을 참조하세요. 이 조사 결과 유형에 대한 세부 정보는 다음과 같습니다.

- 오픈 포트 범위 - EC2 인스턴스에 액세스할 때 사용할 수 있는 포트 범위입니다.
- 오픈 네트워크 경로 - EC2 인스턴스에 대한 오픈 액세스 경로가 표시됩니다. 경로에서 항목을 선택하면 자세한 내용을 확인할 수 있습니다.
- 해결 - 오픈 네트워크 경로를 닫는 방법을 권장합니다.

Amazon Inspector 점수 보기 및 취약성 인텔리전스 세부 정보 이해

Amazon Inspector는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 조사 결과에 대한 점수를 생성합니다. Amazon Inspector 콘솔에서 Amazon Inspector 점수 및 취약성 인텔리전스 세부 정보를 볼 수 있습니다. Amazon Inspector 점수는 공통 취약성 점수 시스템의 지표와 비교할 수 있는 세부 정보를 제공합니다. 이러한 세부 정보는 패키지 취약성 조사 결과에 대해서만 제공됩니다. 이 단원에서는 Amazon Inspector 점수를 해석하고 취약성 인텔리전스 세부 정보를 이해하는 방법에 대해 설명합니다.

Amazon Inspector 점수

Amazon Inspector는 각 Amazon EC2 조사 결과에 대한 점수를 생성합니다. Amazon Inspector는 기본 CVSS 점수 정보를 컴퓨팅 환경에서 수집한 정보(예: 네트워크 연결성 데이터 및 악용 가능성 데이터)와 상호 연관시켜 점수를 결정합니다. Amazon Inspector는 Amazon, Debian 및 RHEL 공급업체를 지원합니다. 각 공급업체는 CVSS v3.1 기본 점수를 제공합니다. 다른 공급업체의 경우 Amazon Inspector는 National Vulnerability Database(NVD)에서 제공하는 CVSS 기본 점수를 사용합니다.

FedRAMP 요구 사항으로 인해 Amazon Inspector는 CVSS v3.1 기본 점수를 기본 점수로 사용합니다. 그러나 CVSS 4.0 기본 점수는 사용 가능한 경우 취약성 메타데이터에 포함됩니다. CVSS 4.0 기본 점수는 취약성 평가를 개선하기 위한 추가 지표를 제공합니다. 결과의 취약성 세부 정보 및 내보낸 조사 결과에서 CVSS 기본 점수의 소스 및 버전을 찾을 수 있습니다.

> **Note**
>
> Ubuntu를 실행하는 Linux 인스턴스에는 Amazon Inspector 점수를 사용할 수 없습니다. Ubuntu는 CVSS 점수와 다른 사용자 지정 심각도 등급 시스템을 사용합니다.

Amazon Inspector 점수 세부 정보

조사 결과의 세부 정보 페이지를 열면 Inspector 점수 및 취약성 인텔리전스 탭을 선택할 수 있습니다. 이 패널에는 기본 점수와 Inspector 점수 간의 차이가 표시됩니다. 이 섹션에서는 Amazon Inspector에서 Amazon Inspector 점수와 소프트웨어 패키지의 공급업체 점수를 조합하여 심각도 등급을 할당하는 방식에 대해 설명합니다. 점수가 서로 다른 경우 이 패널에는 다른 이유에 대한 설명이 표시됩니다.

CVSS 점수 지표 섹션에서는 CVSS 기본 점수 지표와 Inspector 점수를 비교한 표를 볼 수 있습니다. 비교된 지표는 first.org에서 관리하는 CVSS 사양 문서에 정의된 기본 지표입니다. 다음은 기본 지표를 요약한 것입니다.

공격 벡터

취약성이 악용될 수 있는 컨텍스트입니다. Amazon Inspector 조사 결과의 경우 네트워크, 인접 네트워크 또는 로컬일 수 있습니다.

공격 복잡도

공격자가 취약성을 악용할 때 직면하게 될 난이도를 나타냅니다. 낮은 점수는 공격자가 취약성을 악용하기 위해 충족해야 할 추가 조건이 거의 또는 전혀 없다는 것을 의미합니다. 높은 점수는 공격자가 이 취약성을 이용해 공격에 성공하려면 상당한 노력을 투자해야 한다는 것을 의미합니다.

필수 권한

공격자가 취약성을 악용하는 데 필요한 권한 수준을 나타냅니다.

사용자 상호 작용

이 지표는 이 취약성을 이용한 공격이 성공하려면 공격자가 아닌 다른 사람이 필요한지 여부를 나타냅니다.

범위

한 취약한 구성 요소의 취약성이 취약한 구성 요소의 보안 범위를 벗어난 구성 요소의 리소스에 영향을 미치는지 여부를 나타냅니다. 이 값이 변경되지 않음인 경우 영향을 받은 리소스와 영향을 받는 리소스가 동일합니다. 이 값이 변경됨인 경우 취약한 구성 요소를 악용하여 여러 보안 기관에서 관리하는 리소스에 영향을 미칠 수 있습니다.

기밀성

취약성이 악용될 때 리소스 내 데이터의 기밀성에 미치는 영향 수준을 측정합니다. 점수 범위는 기밀성이 손실되지 않는 없음부터 리소스 내의 모든 정보가 공개되거나 암호 또는 암호화 키 등의 기밀 정보가 공개될 수 있는 높음까지 다양합니다.

무결성

취약성이 악용될 경우 영향을 받는 리소스 내의 데이터 무결성에 미치는 영향 수준을 측정합니다. 공격자가 영향을 받는 리소스 내에서 파일을 수정하면 무결성이 훼손됩니다. 점수 범위는 취약성을 악용하더라도 공격자가 정보를 수정할 수 없는 없음부터 취약성을 악용할 경우 공격자가 일부 또는 모든 파일을 수정할 수 있거나 수정할 수 있는 파일이 심각한 결과를 초래할 수 있는 높음까지 다양합니다.

가용성

취약성이 악용될 때 영향을 받는 리소스의 가용성에 미치는 영향 수준을 측정합니다. 점수 범위는 취약성이 가용성에 전혀 영향을 미치지 않는 없음부터 악용될 경우 공격자가 리소스의 가용성을 완전히 거부하거나 서비스를 사용할 수 없게 만들 수 있는 높음까지 다양합니다.

취약성 인텔리전스

이 섹션에서는 Amazon의 CVE에 대한 사용 가능한 인텔리전스와 사이버 보안 및 인프라 보안 기관(CISA)과 같은 업계 표준 보안 인텔리전스 소스를 요약합니다.

> **Note**
>
> CISA 또는 Amazon의 Intel은 일부 CVEs에서 사용할 수 없습니다.

취약성 인텔리전스 세부 정보는 콘솔에서 또는 BatchGetFindingDetails API를 사용하여 볼 수 있습니다. 콘솔에서 다음 세부 정보를 확인할 수 있습니다.

ATT&CK

이 섹션에서는 CVE와 관련된 MITRE 전술, 기법 및 절차(TTP)를 설명합니다. 관련 TTP가 표시되며, 해당하는 TTP가 두 개 이상인 경우 링크를 선택하여 전체 목록을 볼 수 있습니다. 전술 또는 기법을 선택하면 MITRE 웹 사이트에서 해당 전술 또는 기법에 대한 정보가 열립니다.

CISA

이 섹션에서는 취약성과 연관된 관련 날짜를 다룹니다. 사이버 보안 및 인프라 보안국(CISA)에서 적극적인 악용의 증거를 바탕으로, 알려진 악용 취약성 카탈로그에 취약성을 추가한 날짜와 CISA에서 시스템에 패치를 적용할 것으로 예상하는 기한입니다. 이 정보는 CISA에서 제공합니다.

알려진 멀웨어

이 섹션에는 이 취약성을 악용하는 알려진 악용 키트와 도구가 나열되어 있습니다.

최근 보고 시간

이 섹션에는 해당 취약성에 대해 마지막으로 알려진 공개 악용 날짜가 표시됩니다.

Amazon Inspector 조사 결과의 심각도 수준 이해

Amazon Inspector는 조사 결과를 생성할 때 조사 결과에 심각도 등급을 할당합니다. 심각도 등급은 조사 결과를 평가하고 우선순위를 정하는 데 도움이 됩니다. 조사 결과에 대한 심각도 등급은 정보, 낮음, 중간, 높음, 중요 등의 수치 점수 및 수준에 해당합니다. Amazon Inspector는 조사 결과 유형에 따라 조사 결과의 심각도 등급을 결정합니다. 이 섹션에서는 Amazon Inspector가 각 조사 결과 유형에 대한 심각도 등급을 결정하는 방법에 대해 설명합니다.

소프트웨어 패키지 취약성 심각도

Amazon Inspector에서는 소프트웨어 패키지 취약성에 대한 심각도 점수의 기준으로 NVD/CVSS 점수가 사용됩니다. NVD/CVSS 점수는 NVD에서 발표하고 CVSS에서 정의한 취약성 심각도 점수로, 공격

복잡도, 익스플로잇 코드 성숙도, 필요한 권한 등의 보안 지표로 구성됩니다. Amazon Inspector는 취약성의 심각도를 반영하여 1부터 10까지의 숫자로 점수를 산출합니다. 이 점수는 시간이 지나도 일정하게 유지되는 취약성의 본질적인 특성에 따라 취약성의 심각도를 반영하기 때문에 Amazon Inspector에서는 이를 기본 점수로 분류합니다. 또한 이 점수는 배포된 여러 환경 전반에서 합리적인 최악의 영향을 가정한 것입니다. CVSS v3 표준은 CVSS 점수를 다음과 같은 심각도 등급에 매핑합니다.

점수	등급
0	Informational
0.1–3.9	Low
4.0–6.9	Medium
7.0–8.9	High
9.0–10.0	Critical

패키지 취약성 결과의 심각도로 분류되지 않음이 할당될 수도 있습니다. 이는 공급업체가 탐지된 취약성에 대해 취약성 점수를 아직 설정하지 않은 경우입니다. 이 경우 조사 결과의 참조 URL을 사용하여 취약성을 조사하고 그에 따라 대응하는 것이 좋습니다.

패키지 취약성 조사 결과에는 다음과 같은 점수와 관련 점수 벡터가 조사 결과 세부 정보의 일부로 포함됩니다.

- EPSS 점수
- Inspector 점수
- Amazon CVE의 CVSS 3.1
- NVD의 CVSS 3.1
- NVD의 CVSS 2.0(해당하는 경우)

코드 취약성 심각도

코드 취약성 조사 결과의 경우 Amazon Inspector는 해당 조사 결과를 생성한 Amazon Q 탐지기에서 정의한 심각도 수준을 사용합니다. CVSS v3 채점 시스템을 사용하여 각 탐지기에 심각도가 할당됩니다.

네트워크 연결성 심각도

Amazon Inspector는 노출된 서비스, 포트 및 프로토콜과 개방 경로 유형을 기반으로 네트워크 연결성 취약성의 심각도를 결정합니다. 다음 표에는 이러한 심각도 등급이 정의되어 있습니다. 개방 경로 등급 열의 값은 가상 게이트웨이, 피어링된 VPCs 및 AWS Direct Connect 네트워크의 개방 경로를 나타냅니다. 기타 모든 노출된 서비스, 포트 및 프로토콜에는 정보 심각도 등급이 할당됩니다.

서비스:	TCP 포트	UDP 포트	인터넷 경로 등급	개방 경로 등급
DHCP	67, 68, 546, 547	67, 68, 546, 547	Medium	Informational
Elasticsearch	9300, 9200	NA	Medium	Informational
FTP	21	21	High	Medium
Global catalog LDAP	3268	NA	Medium	Informational
Global catalog LDAP over TLS	3269	NA	Medium	Informational
HTTP	80	80	Low	Informational
HTTPS	443	443	Low	Informational
Kerberos	88, 464, 543, 544, 749, 751	88, 464, 749, 750, 751, 752	Medium	Informational
LDAP	389	389	Medium	Informational
LDAP over TLS	636	NA	Medium	Informational
MongoDB	27017, 27018, 27019, 28017	NA	Medium	Informational
MySQL	3306	NA	Medium	Informational
NetBIOS	137, 139	137, 138	Medium	Informational
NFS	111, 2049, 4045, 1110	111, 2049, 4045, 1110	Medium	Informational

Oracle	1521, 1630	NA	Medium	Informational
PostgreSQL	5432	NA	Medium	Informational
Print services	515	NA	High	Medium
RDP	3389	3389	Medium	Low
RPC	111, 135, 530	111, 135, 530	Medium	Informational
SMB	445	445	Medium	Informational
SSH	22	22	Medium	Low
SQL Server	1433	1434	Medium	Informational
Syslog	601	514	Medium	Informational
Telnet	23	23	High	Medium
WINS	1512, 42	1512, 42	Medium	Informational

Amazon Inspector에서 조사 결과 관리

Amazon Inspector를 사용하면 다양한 방식으로 조사 결과를 관리할 수 있습니다. 조사 결과는 상태를 기준으로 필터링할 수 있습니다. 필터 기준에 따라 조사 결과를 검색할 수 있습니다. 조사 결과 목록에서 조사 결과를 제외하는 억제 규칙을 생성할 수 있습니다. 조사 결과를 AWS Security Hub CSPM Amazon EventBridge 및 Amazon Simple Storage Service(Amazon S3)로 내보낼 수도 있습니다.

주제

- Amazon Inspector 조사 결과 필터링
- Amazon Inspector 조사 결과 숨기기
- Amazon Inspector 조사 결과 보고서 내보내기
- Amazon EventBridge를 사용하여 Amazon Inspector 조사 결과에 대한 사용자 지정 응답 생성

Amazon Inspector 조사 결과 필터링

필터 기준을 사용하여 Amazon Inspector 조사 결과를 필터링할 수 있습니다. 조사 결과가 필터 기준과 일치하지 않는 경우 Amazon Inspector는 해당 조사 결과를 보기에서 제외합니다. 이 단원에서는 필터 기준을 사용하여 Amazon Inspector 조사 결과를 필터링하는 방법에 대해 설명합니다.

Amazon Inspector 콘솔에서 필터 생성

각 조사 결과 보기에서 필터 기능을 사용하여 특정한 특성을 가진 조사 결과를 찾을 수 있습니다. 다른 탭 보기로 이동하면 필터는 제거됩니다.

필터는 필터 기준으로 구성되고, 필터 기준은 필터 값과 쌍을 이루는 하나의 필터 속성으로 구성됩니다. 필터 기준과 일치하지 않는 조사 결과는 조사 결과 목록에서 제외됩니다. 예를 들어 관리자 계정과 연결된 모든 조사 결과를 보려면 AWS 계정 ID 속성을 선택하고 12자리 AWS 계정 ID의 값과 페어링하면 됩니다.

모든 조사 결과에 적용되는 필터 기준이 있는 반면, 특정 리소스 유형이나 조사 결과 유형에만 적용되는 필터 기준도 있습니다.

조사 결과 보기에 필터를 적용하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.

2. 탐색 창에서 조사 결과를 선택합니다. 기본 보기에는 활성 상태인 모든 조사 결과가 표시됩니다.
3. 조사 결과를 기준별로 필터링하려면 필터 추가 막대를 선택하여 해당 보기에 적용할 수 있는 모든 필터 기준 목록을 표시합니다. 보기마다 다른 필터 기준을 사용할 수 있습니다.
4. 목록에서 필터 기준을 선택합니다.
5. 기준 입력 창에 원하는 필터 값을 입력하여 해당 기준을 정의합니다.
6. 적용을 선택하여 해당 필터 기준을 현재 결과에 적용합니다. 필터 입력 막대를 다시 선택하여 다른 필터 기준을 계속 추가할 수 있습니다.
7. (선택 사항) 표시되지 않은 조사 결과나 종결된 조사 결과를 보려면 필터 막대에서 활성을 선택한 다음 표시되지 않음 또는 종결됨을 선택합니다. 활성 결과, 표시되지 않은 조사 결과 및 종결된 조사 결과를 동일한 보기에서 보려면 모두 보기를 선택합니다.

Amazon Inspector 조사 결과 숨기기

기준과 일치하는 조사 결과를 숨기도록 억제 규칙을 생성할 수 있습니다. 예를 들어 심각도 등급에 따라 조사 결과를 숨기는 억제 규칙을 생성할 수 있습니다. Amazon Inspector에서 억제 규칙과 일치하는 조사 결과가 생성되면 Amazon Inspector는 해당 조사 결과를 억제하고 표시되지 않도록 숨깁니다. Amazon Inspector는 조사 결과가 해결될 때까지 억제된 조사 결과를 저장합니다. 억제된 조사 결과가 해결되면 Amazon Inspector는 조사 결과를 종결합니다. 억제된 조사 결과는 콘솔에서 확인할 수 있습니다.

가장 중요한 조사 결과의 우선순위를 지정하는 억제 규칙을 생성할 수 있습니다. 억제 규칙은 조사 결과를 보기에서만 숨기므로 조사 결과에 영향을 주지 않습니다. 조사 결과를 종결하거나 수정하는 억제 규칙은 생성할 수 없습니다. Amazon EventBridge 규칙을 AWS Security Hub CSPM 사용하여에서 원치 않는 결과를 억제할 수도 있습니다. 이 단원의 절차에서는 억제 규칙을 생성, 보기, 편집, 삭제하는 방법에 대해 설명합니다.

> **Note**
>
> 조직의 위임된 관리자만 억제 규칙을 생성하고 관리할 수 있습니다.

억제 규칙 생성

억제 규칙을 생성하여 기본적으로 표시되는 조사 결과 목록을 필터링할 수 있습니다. CreateFilter API를 사용하여 SUPPRESS를 `action` 값으로 지정하여 프로그래밍 방식으로 억제 규칙을 생성할 수 있습니다.

> **Note**
>
> 독립형 계정과 Amazon Inspector 위임 관리자만 억제 규칙을 생성하고 관리할 수 있습니다. 조직의 멤버는 탐색 창에서 억제 규칙 옵션을 볼 수 없습니다.

억제 규칙을 생성하려면(콘솔)

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 억제 규칙을 선택합니다. 그런 다음 규칙 생성을 선택합니다.
3. 각 기준에 대해 다음을 수행합니다.
   - 필터 막대를 선택하여 억제 규칙에 추가할 수 있는 필터 기준 목록을 표시합니다.
   - 억제 규칙의 필터 기준을 선택합니다.
4. 기준 추가를 마쳤으면 규칙 이름과 설명(선택 사항)을 입력합니다.
5. 규칙 저장을 선택합니다. Amazon Inspector에서 새 억제 규칙을 즉시 적용하고 기준과 일치하는 조사 결과는 숨깁니다.

숨겨진 조사 결과 보기

기본적으로 Amazon Inspector는 숨겨진 조사 결과를 Amazon Inspector 콘솔에 표시하지 않습니다. 하지만 특정 규칙에 따라 숨겨진 조사 결과를 볼 수는 있습니다.

숨겨진 조사 결과를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 억제 규칙을 선택합니다.
3. 억제 규칙 목록에서 규칙 제목을 선택합니다.

억제 규칙 편집

억제 규칙은 언제든지 변경할 수 있습니다.

억제 규칙을 수정하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 억제 규칙을 선택합니다.
3. 변경하려는 억제 규칙의 이름을 선택한 다음 편집을 선택합니다.
4. 원하는 내용을 변경하고 저장을 선택합니다.

억제 규칙 삭제

억제 규칙을 삭제할 수 있습니다. 억제 규칙을 삭제하면 Amazon Inspector에서 규칙 기준을 충족하고 다른 규칙에 의해 차단되지 않는 새로운 조사 결과와 기존 조사 결과의 억제가 중단됩니다.

억제 규칙을 삭제하면 해당 규칙의 기준을 충족하는 새로운 조사 결과와 기존 조사 결과가 활성 상태가 됩니다. 즉, Amazon Inspector 콘솔에 기본적으로 표시됩니다. 또한 Amazon Inspector는 이러한 결과를 AWS Security Hub CSPM 및 Amazon EventBridge에 이벤트로 게시합니다.

억제 규칙을 삭제하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 억제 규칙을 선택합니다.
3. 삭제하려는 억제 규칙 제목 옆의 확인란을 선택합니다.
4. 삭제를 선택한 다음 선택을 확인하여 규칙을 영구 삭제합니다.

Amazon Inspector 조사 결과 보고서 내보내기

조사 결과 보고서는 조사 결과에 대한 자세한 스냅샷을 제공하는 CSV 또는 JSON 파일입니다. 조사 결과 보고서를 AWS Security Hub CSPM Amazon EventBridge 및 Amazon Simple Storage Service(Amazon S3)로 내보낼 수 있습니다. 조사 결과 보고서를 구성할 때 보고서에 포함할 조사 결과를 지정해야 합니다. 기본적으로 조사 결과 보고서에는 모든 활성 조사 결과에 대한 데이터가 포함됩니다. 조직의 위임된 관리자인 경우 조사 결과 보고서에 조직의 모든 멤버 계정에 대한 데이터가 포함됩니다. 조사 결과 보고서를 사용자 지정하려면 필터를 생성하여 해당 보고서에 적용합니다.

조사 결과 보고서를 내보내면 Amazon Inspector는 AWS KMS key 지정한를 사용하여 조사 결과 데이터를 암호화합니다. Amazon Inspector는 조사 결과 데이터를 암호화한 후 사용자가 지정한 Amazon S3 버킷에 조사 결과 보고서를 저장합니다. AWS KMS 키는 Amazon S3 버킷 AWS 리전 과 동일한

에서 사용해야 합니다. AWS KMS 키 정책은 Amazon Inspector가 이를 사용하도록 허용해야 하며, Amazon S3 버킷 정책은 Amazon Inspector가 객체를 추가하도록 허용해야 합니다. 조사 결과 보고서를 내보낸 후에는 Amazon S3 버킷에서 다운로드하거나 새 위치로 전송할 수 있습니다. Amazon S3 버킷을 내보낸 다른 조사 결과 보고서의 리포지토리로 사용할 수도 있습니다.

이 단원에서는 Amazon Inspector 콘솔에서 조사 결과 보고서를 내보내는 방법에 대해 설명합니다. 다음 작업을 수행하려면 권한을 확인하고, Amazon S3 버킷을 구성하고,를 구성하고 AWS KMS key, 결과 보고서를 구성하고 내보내야 합니다.

> **Note**
>
> Amazon Inspector CreateFindingsReport API를 사용하여 조사 결과 보고서를 내보내는 경우 활성 조사 결과만 볼 수 있습니다. 억제되거나 종결된 조사 결과를 보려면 필터 기준의 일부로 `SUPPRESSED` 또는 `CLOSED`를 지정해야 합니다.

작업

- 1단계: 권한 확인
- 2단계: S3 버킷 구성
- 3단계: 구성 AWS KMS key
- 4단계: 조사 결과 보고서 구성 및 내보내기
- 내보내기 오류 해결

1단계: 권한 확인

> **Note**
>
> 조사 결과 보고서를 처음 내보낸 후에는 1–3단계를 선택적으로 수행할 수 있습니다. 다음 단계는 동일한 Amazon S3 버킷을 사용할지 여부와 내보낸 다른 결과 보고서에 AWS KMS key 사용할지 여부를 기반으로 합니다. 1–3단계를 완료한 후 프로그래밍 방식으로 조사 결과 보고서를 내보내려면Amazon Inspector API의 CreateFindingsReport 작업을 사용합니다.

Amazon Inspector에서 조사 결과 보고서를 내보내기 전에 조사 결과 보고서를 내보내고 보고서를 암호화 및 저장하기 위한 리소스를 구성하는 데 필요한 권한이 있는지 확인합니다. 권한을 확인하려면 AWS Identity and Access Management (IAM)을 사용하여 IAM 자격 증명에 연결된 IAM 정책을 검토합

니다. 그런 다음 해당 정책의 정보를 다음 작업 목록과 비교하여 조사 결과 보고서 내보내기를 위해 사용자가 수행할 수 있어야 하는 작업을 확인합니다.

Amazon Inspector –

Amazon Inspector의 경우 다음 작업을 수행할 수 있는지 확인합니다.

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

이러한 작업을 통해 계정의 조사 결과 데이터를 검색하고 해당 데이터를 결과 보고서로 내보낼 수 있습니다.

대용량 보고서를 프로그래밍 방식으로 내보내려는 경우 `inspector2:GetFindingsReportStatus`(보고서 상태 확인) 및 `inspector2:CancelFindingsReport`(진행 중인 내보내기 취소) 작업을 수행할 수 있는 권한이 있는지도 확인할 수 있습니다.

AWS KMS

에서 다음 작업을 수행할 수 있는지 AWS KMS확인합니다.

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

이러한 작업을 통해 Amazon Inspector에서 보고서를 암호화하는 데 사용할 AWS KMS key 에 대한 키 정책을 검색하고 업데이트할 수 있습니다.

Amazon Inspector 콘솔을 사용하여 보고서를 내보내려면 다음 AWS KMS 작업을 수행할 수 있는지도 확인합니다.

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

이러한 작업을 통해 계정의 AWS KMS keys 에 대한 정보를 검색하고 표시할 수 있습니다. 그런 다음 이러한 키 중 하나를 선택하여 보고서를 암호화할 수 있습니다.

보고서 암호화를 위한 KMS 키를 새로 생성하려는 경우 `kms:CreateKey` 작업을 수행할 수 있어야 합니다.

Amazon S3

Amazon S3의 경우 다음 작업을 수행할 수 있는지 확인합니다.

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

  이러한 작업을 통해 Amazon Inspector에서 보고서를 저장할 S3 버킷을 생성하고 구성할 수 있습니다. 또한 버킷에서 객체를 추가하고 삭제할 수도 있습니다.

  Amazon Inspector 콘솔을 사용하여 보고서를 내보내려는 경우 `s3:ListAllMyBuckets` 및 `s3:GetBucketLocation` 작업을 수행할 수 있는지도 확인합니다. 이러한 작업을 통해 계정의 S3 버킷에 대한 정보를 검색하고 표시할 수 있습니다. 그런 다음 이러한 버킷 중 하나를 선택하여 보고서를 저장할 수 있습니다.

필요한 작업을 하나 이상 수행할 수 없는 경우 다음 단계로 진행하기 전에 AWS 관리자에게 도움을 요청하세요.

2단계: S3 버킷 구성

권한을 확인했으면 조사 결과 보고서를 저장할 S3 버킷을 구성할 준비가 된 것입니다. 자체 계정의 기존 버킷이거나 다른이 소유하고 AWS 계정 있으며 사용자가 액세스할 수 있는 기존 버킷일 수 있습니다. 보고서를 새 버킷에 저장하려면 진행하기 전에 버킷을 생성합니다.

S3 버킷은 내보내려는 AWS 리전 결과 데이터와 동일한에 있어야 합니다. 예를 들어, Amazon Inspector를 미국 동부(버지니아 북부) 리전에서 사용 중이고 해당 리전에 대한 조사 결과 데이터를 내보내려면 버킷도 미국 동부(버지니아 북부) 리전에 있어야 합니다.

또한 버킷 정책에서 Amazon Inspector가 버킷에 객체를 추가할 수 있도록 허용해야 합니다. 이 주제에서는 버킷 정책을 업데이트하는 방법을 설명하고 정책에 추가할 명령문의 예를 제공합니다. 버킷 정책 추가 및 업데이트에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서에서 버킷 정책 사용을 참조하세요.

다른 계정이 소유하고 있는 S3 버킷에 보고서를 저장하려면 버킷 소유자와 협력하여 버킷 정책을 업데이트합니다. 버킷 URI도 확보합니다. 보고서를 내보낼 때 이 URI를 입력해야 합니다.

버킷 정책을 업데이트하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon S3 콘솔(https://console.aws.amazon.com/s3)을 엽니다.
2. 탐색 창에서 버킷을 선택합니다.
3. 조사 결과 보고서를 저장할 S3 버킷을 선택합니다.
4. 권한 탭을 선택합니다.
5. 버킷 정책 섹션에서 편집을 선택합니다.
6. 다음 예제 명령문을 클립보드로 복사합니다.

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. Amazon S3 콘솔의 버킷 정책 편집기에서 위의 명령문을 정책에 붙여 넣어 정책에 추가합니다.

   명령문을 추가할 때 구문이 올바른지 확인합니다. 버킷 정책에는 JSON 형식이 사용됩니다. 즉, 정책에 명령문을 추가하는 위치에 따라 명령문 앞이나 뒤에 쉼표를 추가해야 합니다. 명령문을 마지막 명령문으로 추가하는 경우 위 명령문의 닫는 괄호 뒤에 쉼표를 추가합니다. 명령문을 첫 번째 명령문으로 추가하거나 기존 두 명령문 사이에 추가하는 경우 명령문의 닫는 괄호 뒤에 쉼표를 추가합니다.

8. 사용 환경에 적합한 값으로 명령문을 업데이트합니다.

   - *amzn-s3-demo-bucket*은 버킷 이름입니다.
   - *111122223333*은 AWS 계정의 계정 ID입니다.
   - *##*은 Amazon Inspector를 사용 중이며 Amazon Inspector AWS 리전 가 버킷에 보고서를 추가하도록 허용하려는 입니다. 예를 들어 미국 동부(버지니아 북부) 리전의 경우 `us-east-1`입니다.

   > **Note**
   >
   > 수동으로 활성화된에서 Amazon Inspector를 사용하는 경우 `Service` 필드 값에 적절한 리전 코드 AWS 리전도 추가합니다. 이 필드는 Amazon Inspector 서비스 위탁자를 지정합니다.
   > 예를 들어 리전 코드가 `me-south-1`인 중동(바레인) 리전에서 Amazon Inspector를 사용하는 경우, 명령문에서 `inspector2.amazonaws.com`을 `inspector2.me-south-1.amazonaws.com`으로 바꿉니다.

   예제 명령문에는 다음과 같은 두 개의 IAM 전역 조건 키를 사용하는 조건이 정의되어 있습니다.

   - AWS:sourceAccount - 이 조건을 사용하면 Amazon Inspector에서 사용자의 계정에 대해서만 버킷에 보고서를 추가하고, Amazon Inspector가 다른 계정에 대해 버킷에 보고서를 추가하지 못하도록 합니다. 더 구체적으로, 이 조건은 `aws:SourceArn` 조건에 지정된 리소스 및 작업에 대해 버킷을 사용할 수 있는 계정을 지정합니다.

     버킷의 추가 계정에 대한 보고서를 저장하려면 각 추가 계정의 계정 ID를 이 조건에 추가합니다. 예제:

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

- AWS:sourceARN - 이 조건은 버킷에 추가되는 객체의 소스에 따라 버킷에 대한 액세스를 제한하고, 다른이 버킷 AWS 서비스 에 객체를 추가하지 못하도록 합니다. 또한 사용자 계정에 대해 다른 작업을 수행하는 동안 Amazon Inspector가 버킷에 객체를 추가하지 못하도록 합니다. 더 구체적으로, 이 조건은 객체가 조사 결과 보고서인 경우에만, 그리고 해당 보고서가 조건에 지정된 계정과 리전에서 생성된 경우에만 Amazon Inspector가 버킷에 객체를 추가할 수 있도록 허용합니다.

  Amazon Inspector가 추가 계정에 대해 지정된 작업을 수행할 수 있도록 하려면 이 조건에 각 추가 계정의 Amazon 리소스 이름(ARN)을 추가합니다. 예제:

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

  aws:SourceAccount 및 aws:SourceArn 조건에 지정된 계정이 일치해야 합니다.

  두 조건 모두 Amazon S3와의 트랜잭션 중에 Amazon Inspector가 혼동되는 대리자로 사용되는 것을 방지하는 데 도움이 됩니다. 권장하지는 않지만 버킷 정책에서 이러한 조건을 제거할 수 있습니다.

9. 버킷 정책 업데이트가 완료되면 변경 사항 저장을 선택합니다.

3단계: 구성 AWS KMS key

권한을 확인하고 S3 버킷을 구성한 후에는 Amazon Inspector에서 조사 결과 보고서를 암호화하는 데 사용할 AWS KMS key 를 결정해야 합니다. 이 키는 고객 관리형 대칭 암호화 KMS 키여야 합니다. 또한 키는 보고서를 저장하도록 구성한 S3 버킷 AWS 리전 과 동일한에 있어야 합니다.

이 키는 내 계정의 기존 KMS 키이거나 다른 계정에서 소유하고 있는 기존 KMS 키일 수 있습니다. 새 KMS 키를 사용하려면 진행하기 전에 키를 생성합니다. 다른 계정에서 소유하고 있는 기존 키를 사용하려면 키의 Amazon 리소스 이름(ARN)을 확보합니다. Amazon Inspector에서 보고서를 내보낼 때 이 ARN을 입력해야 합니다. KMS 키 설정 생성 및 검토에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서에서 키 관리를 참조하세요.

사용하려는 KMS 키를 결정한 후에는 Amazon Inspector에 키 사용 권한을 부여합니다. 그렇지 않으면 Amazon Inspector에서 보고서를 암호화하고 내보낼 수 없습니다. Amazon Inspector에 키 사용 권한을

부여하려면 키에 대한 키 정책을 업데이트합니다. 키 정책 및 KMS 키 액세스 관리에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서에서 AWS KMS의 키 정책을 참조하세요.

> **Note**
>
> 다음 절차는 Amazon Inspector에서 사용할 수 있도록 기존 키를 업데이트하는 절차입니다. 기존 키가 없는 경우 AWS Key Management Service 개발자 안내서에서 키 생성을 참조하세요.

키 정책을 업데이트하려면

1. 자격 증명을 사용하여 로그인한 다음 https://console.aws.amazon.com/kms AWS KMS 콘솔을 엽니다.
2. 탐색 창에서 고객 관리형 키를 선택합니다.
3. 보고서를 암호화하는 데 사용할 KMS 키를 선택합니다. 이 키는 대칭 암호화(SYMMETRIC_DEFAULT) 키여야 합니다.
4. 키 정책 탭에서 편집을 선택합니다. 키 정책에서 편집 버튼이 보이지 않으면 먼저 정책 보기로 전환을 선택해야 합니다.
5. 다음 예제 명령문을 클립보드로 복사합니다.

```
{
    "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. AWS KMS 콘솔의 키 정책 편집기에서 앞의 문을 키 정책에 붙여 넣어 정책에 추가합니다.

   정책에 성명문을 추가할 때 구문이 올바른지 확인합니다. 키 정책에는 JSON 형식이 사용됩니다. 즉, 정책에 명령문을 추가하는 위치에 따라 명령문 앞이나 뒤에 쉼표를 추가해야 합니다. 명령문을 마지막 명령문으로 추가하는 경우 위 명령문의 닫는 괄호 뒤에 쉼표를 추가합니다. 명령문을 첫 번째 명령문으로 추가하거나 기존 두 명령문 사이에 추가하는 경우 명령문의 닫는 괄호 뒤에 쉼표를 추가합니다.

7. 사용 환경에 적합한 값으로 명령문을 업데이트합니다.

   - *111122223333*은 AWS 계정의 계정 ID입니다.
   - *##*은 Amazon Inspector AWS 리전 가 키로 보고서를 암호화하도록 허용할 입니다. 예를 들어 미국 동부(버지니아 북부) 리전의 경우 `us-east-1`입니다.

   > **Note**
   >
   > 수동으로 활성화된에서 Amazon Inspector를 사용하는 경우 `Service` 필드 값에 적절한 리전 코드 AWS 리전도 추가합니다. 예를 들어 중동(바레인) 리전에서 Amazon Inspector를 사용할 경우 `inspector2.amazonaws.com`을 `inspector2.me-south-1.amazonaws.com`으로 바꿉니다.

   이전 단계의 버킷 정책 예제 명령문과 마찬가지로, 이 예제의 `Condition` 필드는 두 개의 IAM 전역 조건 키를 사용합니다.

   - aws:SourceAccount - 이 조건을 사용하면 Amazon Inspector에서 사용자 계정에 대해서만 지정된 작업을 수행할 수 있습니다. 더 구체적으로, 이 조건은 `aws:SourceArn` 조건에 지정된 리소스 및 작업에 대해 지정된 작업을 수행할 수 있는 계정을 결정합니다.

     Amazon Inspector가 추가 계정에 대해 지정된 작업을 수행할 수 있도록 하려면 이 조건에 각 추가 계정의 계정 ID를 추가합니다. 예제:

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

   - aws:SourceArn - 이 조건은 다른 AWS 서비스 에서 지정된 작업을 수행하지 못하도록 합니다. 또한 사용자 계정에 대해 다른 작업을 수행하는 동안 Amazon Inspector가 키를 사용하지 못하도록 합니다. 즉, 객체가 조사 결과 보고서인 경우에만, 그리고 해당 보고서가 조건에 지정된 계

정과 리전에서 생성된 경우에만 Amazon Inspector가 해당 키로 S3 객체를 암호화할 수 있도록 허용합니다.

Amazon Inspector가 추가 계정에 대해 지정된 작업을 수행할 수 있도록 하려면 이 조건에 각 추가 계정의 ARN을 추가합니다. 예제:

```
"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/*"
]
```

`aws:SourceAccount` 및 `aws:SourceArn` 조건에 지정된 계정이 일치해야 합니다.

이러한 조건은 Amazon Inspector가 트랜잭션 중에 혼동된 대리자로 사용되는 것을 방지하는 데 도움이 됩니다 AWS KMS. 권장하지는 않지만 명령문에서 이러한 조건을 제거할 수 있습니다.

8. 키 정책 업데이트가 완료되면 변경 사항 저장을 선택합니다.

4단계: 조사 결과 보고서 구성 및 내보내기

> **Note**
>
> 조사 결과 보고서는 한 번에 하나만 내보낼 수 있습니다. 현재 내보내기가 진행 중인 경우 내보내기가 완료될 때까지 기다렸다가 다른 조사 결과 보고서를 내보내야 합니다.

권한을 확인하고 조사 결과 보고서를 암호화하고 저장하도록 리소스를 구성했으면 보고서를 구성하고 내보낼 준비가 된 것입니다.

조사 결과 보고서를 구성하고 내보내려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창의 조사 결과에서 모든 조사 결과를 선택합니다.
3. (선택 사항) 조사 결과 테이블 위의 필터 막대를 사용하여 보고서에 포함할 조사 결과를 지정하는 필터 기준을 추가합니다. 기준을 추가하면 Amazon Inspector에서 기준과 일치하는 조사 결과만

포함하도록 테이블을 업데이트합니다. 이 테이블은 보고서에 포함될 데이터의 미리 보기를 제공합니다.

> **Note**
>
> 필터 기준을 추가하는 것이 좋습니다. 그렇지 않으면 현재에서 활성 상태 AWS 리전 인 모든 결과에 대한 데이터가 보고서에 포함됩니다. 조직의 Amazon Inspector 관리자인 경우 여기에 조직 내 모든 멤버 계정에 대한 조사 결과 데이터가 포함됩니다.
> 보고서에 전체 또는 여러 조사 결과의 데이터가 포함되어 있는 경우 보고서를 생성하고 내보내는 데 시간이 오래 걸릴 수 있으며, 한 번에 하나의 보고서만 내보낼 수 있습니다.

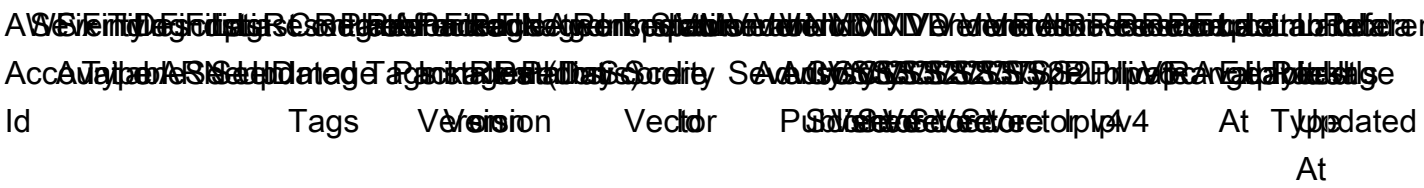
4. 조사 결과 내보내기를 선택합니다.
5. 내보내기 설정 섹션의 내보내기 파일 유형에서 보고서의 파일 형식을 지정합니다.

   - 데이터가 포함된 JavaScript Object Notation(.json) 파일을 생성하려면 JSON을 선택합니다.

     JSON 옵션을 선택하면 보고서에 각 조사 결과에 대한 모든 필드가 포함됩니다. 가능한 JSON 필드 목록은 Amazon Inspector API 참조에서 조사 결과 데이터 유형을 참조하세요.

   - 데이터가 포함된 쉼표로 구분된 값 (.csv) 파일을 생성하려면 CSV를 선택합니다.

     CSV 옵션을 선택하면 각 조사 결과에 대한 일부 필드, 즉 조사 결과의 주요 속성을 보고하는 약 45개 필드만 보고서에 포함됩니다. 이 필드에는 조사 결과 유형, 제목, 심각도, 상태, 설명, 처음 발견 날짜, 최종 발견 날짜, 수정 버전 있음, AWS 계정 ID, 리소스 ID, 리소스 태그, 해결 등이 있습니다. 여기에는 점수 세부 정보 및 각 조사 결과에 대한 참조 URL을 캡처하는 필드도 포함됩니다. 다음은 조사 결과 보고서의 CSV 헤더 샘플입니다.



6. 내보내기 위치의 S3 URI에서 보고서를 저장할 S3 버킷을 지정합니다.

   - 계정이 소유한 버킷에 보고서를 저장하려면 S3 찾아보기를 선택합니다. Amazon Inspector에서 계정의 S3 버킷 테이블을 표시합니다. 원하는 버킷의 행을 선택한 다음 선택을 선택합니다.

> **Tip**
>
> 또한 보고서에 대해 Amazon S3 경로 접두사를 지정하려면 슬래시(/)와 접두사를 S3 URI 상자의 값에 추가합니다. 그러면 Amazon Inspector가 보고서를 버킷에 추가할 때 해당 접두사가 포함되고, Amazon S3는 접두사로 지정된 경로를 생성합니다.
> 예를 들어 AWS 계정 ID를 접두사로 사용하고 계정 ID가 111122223333인 경우 S3 URI 상자**/111122223333**의 값에를 추가합니다.
> 접두사는 S3 버킷 내에서 디렉터리 경로와 비슷합니다. 유사한 파일을 파일 시스템의 폴더에 함께 저장하는 것처럼 유사한 객체를 버킷에 그룹화할 수 있습니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서에서 Amazon S3 콘솔에서 폴더를 사용하여 객체 구성을 참조하세요.

- 다른 계정이 소유한 버킷에 보고서를 저장하려면 버킷 URI를 입력합니다. 예를 들어 **s3://DOC-EXAMPLE_BUCKET**과 같이 입력합니다. 여기서 DOC-EXAMPLE_BUCKET은 버킷의 이름입니다. 버킷 소유자가 버킷 속성에서 이 정보를 찾을 수 있습니다.

7. KMS 키에서 보고서를 암호화하는 데 AWS KMS key 사용할를 지정합니다.

   - 내 계정의 키를 사용하려면 목록에서 키를 선택합니다. 목록에 계정의 고객 관리형 대칭 암호화 KMS 키가 표시됩니다.
   - 다른 계정이 소유한 키를 사용하려면 키의 Amazon 리소스 이름(ARN)을 입력합니다. 키 소유자가 키 속성에서 이 정보를 찾을 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 키 ID 및 ARN 찾기를 참조하세요.

8. 내보내기를 선택합니다.

Amazon Inspector에서 조사 결과 보고서를 생성하고 지정한 KMS 키로 암호화한 다음 지정한 S3 버킷에 추가합니다. 보고서에 포함되도록 선택한 조사 결과 수에 따라 이 프로세스는 몇 분 또는 몇 시간이 걸릴 수 있습니다. 내보내기가 완료되면 Amazon Inspector에서 조사 결과 보고서를 성공적으로 내보냈다는 메시지를 표시합니다. 선택적으로 메시지에서 보고서 보기를 선택하여 Amazon S3의 보고서로 이동합니다.

보고서는 한 번에 하나만 내보낼 수 있습니다. 내보내기가 현재 진행 중이라면 내보내기가 완료될 때까지 기다린 후 다른 보고서를 내보냅니다.

내보내기 오류 해결

조사 결과 보고서를 내보내려고 할 때 오류가 발생하면 Amazon Inspector에서 오류를 설명하는 메시지를 표시합니다. 이 주제에 설명된 정보를 참고하여 오류의 가능한 원인과 해결 방법을 파악할 수 있습니다.

예를 들어 S3 버킷이 현재에 AWS 리전 있고 버킷의 정책에서 Amazon Inspector가 버킷에 객체를 추가하도록 허용하는지 확인합니다. 또한 현재 리전에서 AWS KMS key 가 활성화되어 있는지 확인하고 키 정책에서 Amazon Inspector가 키를 사용하도록 허용하는지 확인합니다.

오류를 해결한 후 보고서를 다시 내보냅니다.

보고서가 여러 개일 수 없음 오류

보고서를 생성하려고 하는데 Amazon Inspector에서 이미 보고서를 생성하고 있는 경우 원인: 진행 중인 보고서가 여러 개일 수 없음이라는 오류 메시지가 나타납니다. Amazon Inspector에서는 계정에 대해 한 번에 하나의 보고서만 생성할 수 있기 때문에 이 오류가 발생합니다.

이 오류를 해결하려면 다른 보고서가 완료될 때까지 기다리거나 새 보고서를 요청하기 전에 보고서를 취소하면 됩니다.

GetFindingsReportStatus 작업을 사용하여 보고서 상태를 확인할 수 있습니다. 이 작업은 현재 생성 중인 보고서의 보고서 ID를 반환합니다.

필요한 경우, `GetFindingsReportStatus` 작업에서 제공한 보고서 ID를 사용하여 현재 진행 중인 내보내기를 CancelFindingsReport 작업을 통해 취소할 수 있습니다.

Amazon EventBridge를 사용하여 Amazon Inspector 조사 결과에 대한 사용자 지정 응답 생성

Amazon Inspector는 새로 생성된 조사 결과와 집계된 조사 결과에 대해 Amazon EventBridge에서 이벤트를 생성합니다. 또한 Amazon Inspector는 조사 결과의 상태 변경에 대한 이벤트도 생성합니다. 즉, 리소스를 다시 시작하거나 리소스와 관련된 태그를 변경하는 등의 작업을 수행하면 Amazon Inspector에서 조사 결과에 대한 새 이벤트를 생성합니다. Amazon Inspector에서 업데이트된 조사 결과에 대해 새 이벤트를 생성할 때 조사 결과 `id`는 동일하게 유지됩니다.

> **Note**
>
> 계정이 Amazon Inspector 위임된 관리자 계정인 경우 EventBridge는 이벤트가 발생한 사용자의 계정과 멤버 계정에 이벤트를 게시합니다.

Amazon Inspector와 함께 EventBridge 이벤트를 사용하면 조사 결과로 드러난 보안 문제에 대응하는 데 도움이 되는 작업을 자동화할 수 있습니다. EventBridge 이벤트에 따라 Amazon Inspector 조사 결과에 대한 알림을 받으려면 EventBridge 규칙을 생성하고 Amazon Inspector의 대상을 지정해야 합니다. EventBridge 규칙을 사용하면 EventBridge가 Amazon Inspector 조사 결과에 대한 알림을 전송할 수 있으며, 대상에는 알림을 전송할 위치를 지정합니다.

Amazon Inspector는 현재 Amazon Inspector를 사용 중인 AWS 리전 의 기본 이벤트 버스로 이벤트를 내보냅니다. 즉, Amazon Inspector AWS 리전 를 활성화하고 EventBridge 이벤트를 수신하도록 Amazon Inspector를 구성한 각에 대한 이벤트 규칙을 구성해야 합니다. Amazon Inspector는 최선의 방식으로 이벤트를 생성합니다.

이 단원에서는 이벤트 스키마의 예를 제공하고 EventBridge 규칙을 생성하는 방법에 대해 설명합니다.

이벤트 스키마

다음은 EC2 조사 결과 이벤트에 대한 Amazon Inspector 이벤트 형식의 예입니다. 다른 조사 결과 유형 및 이벤트 유형의 스키마 예는 EventBridge 스키마를 참조하세요.


```
{
    "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"
    }
}
```


Amazon Inspector 조사 결과를 알리는 EventBridge 규칙 생성

Amazon Inspector 조사 결과의 가시성을 높이기 위해 EventBridge를 사용하여 자동 조사 결과 알림을 메시징 허브로 보내도록 설정할 수 있습니다. 이 주제에서는 `CRITICAL` 및 `HIGH` 심각도 조사 결과에 대한 알림을 이메일, Slack 또는 Amazon Chime으로 보내는 방법에 대해 설명합니다. Amazon Simple Notification Service 주제를 설정한 다음 해당 주제를 EventBridge 이벤트 규칙에 연결하는 방법을 알아봅니다.

1단계. Amazon SNS 주제 및 엔드포인트 설정

자동 알림을 설정하려면 먼저 Amazon Simple Notification Service에서 주제를 설정하고 엔드포인트를 추가해야 합니다. 자세한 내용은 SNS 설명서를 참조하세요.

이 절차는 Amazon Inspector 조사 결과 데이터를 보낼 위치를 설정합니다. SNS 주제는 이벤트 규칙을 생성하는 동안 또는 생성한 후에 EventBridge 이벤트 규칙에 추가할 수 있습니다.

Email setup

SNS 주제 생성

1. Amazon SNS 콘솔(https://console.aws.amazon.com/sns/v3/home)에 로그인합니다.
2. 탐색 창에서 주제를 선택한 다음 주제 생성을 선택합니다.
3. 주제 생성 섹션에서 표준을 선택합니다. 주제 이름을 입력합니다(예: **`Inspector_to_Email`**). 기타 세부 정보는 선택 사항입니다.
4. 주제 생성을 선택합니다. 그러면 새 주제에 대한 세부 정보가 포함된 새 패널이 열립니다.
5. 구독 섹션에서 구독 생성을 선택합니다.
6. a. 프로토콜 메뉴에서 이메일을 선택합니다.
   b. 엔드포인트 필드에 알림을 받을 이메일 주소를 입력합니다.

   > **Note**
   > 구독을 생성한 후 이메일 클라이언트를 통해 구독을 확인해야 합니다.

   c. 구독 생성을 선택합니다.
7. 받은 편지함에서 구독 메시지를 확인하고 구독 확인을 선택합니다.

Slack setup

SNS 주제 생성

1. Amazon SNS 콘솔(https://console.aws.amazon.com/sns/v3/home)에 로그인합니다.
2. 탐색 창에서 주제를 선택한 다음 주제 생성을 선택합니다.
3. 주제 생성 섹션에서 표준을 선택합니다. 주제 이름을 입력합니다(예: **`Inspector_to_Slack`**). 기타 세부 정보는 선택 사항입니다. 주제 생성을 선택하여 엔드포인트 생성을 완료합니다.

채팅 애플리케이션 클라이언트 내 Amazon Q Developer 구성

1. https://console.aws.amazon.com/chatbot/의 채팅 애플리케이션 콘솔에서 Amazon Q Developer로 이동합니다.
2. 구성된 클라이언트 창에서 새 클라이언트 구성을 선택합니다.
3. Slack을 선택한 다음 구성을 선택하여 확인합니다.

> **Note**
>
> Slack을 선택할 때는 허용을 선택하여 채팅 애플리케이션의 Amazon Q Developer이 채널에 액세스할 수 있는 권한을 확인합니다.

4. 새 채널 구성을 선택하여 구성 세부 정보 창을 엽니다.
   a. 채널 이름을 입력합니다.
   b. Slack 채널에서 사용할 채널을 선택합니다.
   c. Slack에서 채널 이름을 마우스 오른쪽 버튼으로 클릭하고 링크 복사를 선택하여 프라이빗 채널의 채널 ID를 복사합니다.
   d. 의 채팅 애플리케이션 내 AWS Management Console Amazon Q Developer 창에서 Slack에서 복사한 채널 ID를 프라이빗 채널 ID 필드에 붙여 넣습니다.
   e. 아직 역할이 없는 경우 권한에서 템플릿을 사용하여 IAM 역할을 생성하도록 선택합니다.
   f. 정책 템플릿에서 알림 권한을 선택합니다. 채팅 애플리케이션의 Amazon Q Developer에 대한 IAM 정책 템플릿입니다. 이 정책은 CloudWatch 경보, 이벤트, 로그, Amazon SNS 주제에 필요한 읽기 및 나열 권한을 제공합니다.
   g. 채널 가드레일 정책으로 AmazonInspector2ReadOnlyAccess를 선택합니다.
   h. 이전에 SNS 주제를 생성할 때 사용한 리전을 선택한 다음 생성한 Amazon SNS 주제를 선택하여 Slack 채널에 알림을 보냅니다.
5. 구성을 선택합니다.

Amazon Chime setup

SNS 주제 생성

1. Amazon SNS 콘솔(https://console.aws.amazon.com/sns/v3/home)에 로그인합니다.
2. 탐색 창에서 주제를 선택한 다음 주제 생성을 선택합니다.

3. 주제 생성 섹션에서 표준을 선택합니다. 주제 이름을 입력합니다(예: **`Inspector_to_Chime`**). 기타 세부 정보는 선택 사항입니다. 주제 생성을 선택하여 완료합니다.

채팅 애플리케이션 클라이언트 내 Amazon Q Developer 구성

1. https://console.aws.amazon.com/chatbot/의 채팅 애플리케이션 콘솔에서 Amazon Q Developer로 이동합니다.
2. 구성된 클라이언트 패널에서 새 클라이언트 구성을 선택합니다.
3. Chime을 선택한 다음 구성을 선택하여 확인합니다.
4. 구성 세부 정보 창에서 채널 이름을 입력합니다.
5. Amazon Chime에서 원하는 채팅룸을 엽니다.
   a. 오른쪽 상단 모서리에 있는 기어 모양 아이콘을 선택하고 Manage webhooks(Webhook 관리)를 선택합니다.
   b. URL 복사를 선택하여 웹후크 URL을 클립보드에 복사합니다.
6. 의 채팅 애플리케이션 내 AWS Management Console Amazon Q Developer 창에서 복사한 URL을 Webhook URL 필드에 붙여 넣습니다.
7. 아직 역할이 없는 경우 권한에서 템플릿을 사용하여 IAM 역할을 생성하도록 선택합니다.
8. 정책 템플릿에서 알림 권한을 선택합니다. 채팅 애플리케이션의 Amazon Q Developer에 대한 IAM 정책 템플릿입니다. CloudWatch 경보, 이벤트, 로그, Amazon SNS 주제에 필요한 읽기 및 나열 권한을 제공합니다.
9. 이전에 SNS 주제를 생성할 때 사용한 리전을 선택한 다음 생성한 Amazon SNS 주제를 선택하여 Amazon Chime 룸에 알림을 보냅니다.
10. 구성을 선택합니다.

2단계. Amazon Inspector 조사 결과에 대한 EventBridge 규칙 생성

1. 자격 증명을 사용하여 로그인합니다.
2. Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)을 엽니다.
3. 탐색 창에서 규칙을 선택한 다음 규칙 생성을 선택합니다.
4. 규칙의 이름과 설명(선택 사항)을 입력합니다.
5. 이벤트 패턴이 있는 규칙을 선택한 후다음을 선택합니다.
6. 이벤트 패턴 창에서 사용자 지정 패턴(JSON 편집기)을 선택합니다.

7. 다음 JSON을 편집기에 붙여 넣습니다.

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

> **Note**
>
> 이 패턴은 Amazon Inspector에서 탐지한 모든 활성 `CRITICAL` 또는 `HIGH` 심각도 조사 결과에 대해 알림을 보냅니다.

이벤트 패턴 입력을 완료하면 다음을 선택합니다.

8. 대상 선택 페이지에서 AWS 서비스를 선택합니다. 그런 다음 대상 유형 선택에서 SNS 주제를 선택합니다.
9. 주제에서 1단계에서 생성한 SNS 주제의 이름을 선택합니다. 다음을 선택합니다.
10. 필요한 경우 선택적 태그를 추가하고 다음을 선택합니다.
11. 규칙을 검토한 다음 규칙 생성을 선택합니다.

Amazon Inspector 다중 계정 환경용 EventBridge

Amazon Inspector 위임 관리자인 경우 멤버 계정의 해당 조사 결과에 따라 EventBridge 규칙이 계정에 표시됩니다. 이전 섹션에 설명된 대로, 관리자 계정의 EventBridge를 통해 조사 결과 알림을 설정하면 다중 계정에 대한 알림을 받게 됩니다. 즉, 내 계정에서 생성된 조사 결과와 이벤트 외에도 멤버 계정에서 생성된 조사 결과와 이벤트에 대한 알림을 받게 됩니다.

조사 결과의 JSON 세부 정보에 있는 `accountId`를 사용하여 Amazon Inspector 조사 결과가 발생한 멤버 계정을 식별할 수 있습니다.

Amazon Inspector에서 대시보드 작업

대시보드는 Amazon Inspector에서 스캔한 리소스에 대해 집계된 통계의 스냅샷을 제공합니다. 대시보드를 사용하여 사용 중인 환경의 적용 범위와 중요한 조사 결과에 대해 확인할 수 있습니다.

> **Note**
>
> 계정이 조직의 위임된 관리자 계정인 경우 대시보드에는 내 계정과 조직의 다른 모든 계정에 대한 정보가 표시됩니다.

이 섹션에서는 대시보드를 보는 방법과 대시보드를 구성하는 구성 요소를 이해하는 방법에 대해 설명합니다.

주제

- 대시보드 보기
- 대시보드 구성 요소 이해 및 데이터 해석

대시보드 보기

대시보드에는 환경 적용 범위에 대한 개요와 중요한 조사 결과가 표시됩니다. 대시보드는 5분마다 데이터를 자동으로 새로 고칩니다. 화면 오른쪽 상단 모서리 근처에 있는 새로 고침 아이콘을 선택하여 데이터를 수동으로 새로 고칠 수 있습니다. 항목을 선택하여 항목에 대한 지원 데이터를 볼 수 있습니다.

> **Note**
>
> 계정이 조직의 위임된 관리자 계정인 경우 계정 필드에 멤버 계정 ID를 입력하여 멤버 계정에 대해 집계된 통계를 볼 수 있습니다.

대시보드를 보려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.

대시보드 구성 요소 이해 및 데이터 해석

대시보드의 각 섹션은 주요 지표와 조사 결과 데이터에 대한 정보를 제공하므로 현재 AWS 리전에서 AWS 리소스의 취약성 상태를 파악할 수 있습니다.

환경 적용 범위

환경 적용 범위 섹션은 Amazon Inspector에서 스캔한 리소스에 대한 통계를 제공합니다. 이 섹션에서는 Amazon Inspector에서 스캔한 Amazon EC2 인스턴스, Amazon ECR 이미지 및 AWS Lambda 함수의 개수와 비율을 확인할 수 있습니다. Amazon Inspector 위임 관리자로서 AWS Organizations를 통해 여러 계정을 관리하는 경우 총 조직 계정 수, Amazon Inspector가 활성화된 횟수 및 해당 조직의 적용 범위 비율도 확인할 수 있습니다. 또한 이 섹션에서는 Amazon Inspector가 적용되지 않는 리소스를 확인할 수도 있습니다. 이러한 리소스에는 조직을 위험에 빠뜨리는 데 악용될 수 있는 취약성이 포함되어 있을 수 있습니다. 자세한 내용은 AWS 환경의 Amazon Inspector 적용 범위 평가을 참조하세요.

적용 범위 그룹을 선택하면 선택한 그룹에 대한 계정 관리 페이지로 이동합니다. 계정 관리 페이지에는 Amazon Inspector가 적용되는 계정, Amazon EC2 인스턴스 및 Amazon ECR 리포지토리에 대한 세부 정보가 표시됩니다.

사용 가능한 적용 범위 그룹은 다음과 같습니다.

- 계정
- 인스턴스
- 컨테이너 리포지토리
- 컨테이너 이미지
- Lambda

중요 조사 결과

중요 조사 결과 섹션에서는 환경의 중요 취약점 수와 환경 내 모든 조사 결과의 총 개수를 확인할 수 있습니다. 이 섹션에서는 리소스 및 평가 유형별로 개수가 표시됩니다. 중요 조사 결과 및 Amazon Inspector의 중요도 결정 방식에 대한 자세한 내용은 Amazon Inspector 조사 결과 이해 섹션을 참조하세요.

중요 조사 결과 그룹을 선택하면 모든 조사 결과 페이지로 이동하며 선택한 그룹과 일치하는 모든 중요 조사 결과를 표시하도록 필터가 자동으로 적용됩니다.

사용 가능한 중요 조사 결과 그룹은 다음과 같습니다.

- Amazon Inspector 코드 스캔 조사 결과

- Amazon EC2 인스턴스 조사 결과
- Amazon ECR 컨테이너 이미지 조사 결과
- Lambda 함수 조사 결과

위험에 기반한 해결 방법

위험에 기반한 해결 방법 섹션에는 사용 환경에서 가장 많은 리소스에 영향을 미치는 중요한 취약점이 있는 상위 5개 소프트웨어 패키지가 표시됩니다. 이러한 패키지를 해결하면 환경에 대한 심각한 위험 수를 크게 줄일 수 있습니다. 소프트웨어 패키지 이름을 선택하면 관련 취약성 세부 정보와 영향을 받는 리소스를 확인할 수 있습니다.

가장 중요한 조사 결과가 있는 계정

가장 중요한 조사 결과가 있는 계정 섹션에는 사용 환경에서 가장 중요한 조사 결과가 있는 상위 5개 AWS 계정과 해당 계정에 대한 총 조사 결과 수가 표시됩니다. AWS Organizations에서 다중 계정 스캔을 수행하도록 Amazon Inspector가 구성된 경우 이 섹션은 위임 관리자 계정에서만 볼 수 있습니다. 이 보기는 위임 관리자가 조직 내에서 가장 위험할 수 있는 계정을 파악하는 데 도움이 됩니다.

계정 ID를 선택하면 영향을 받는 멤버 계정에 대한 자세한 정보를 볼 수 있습니다.

가장 중요한 조사 결과가 있는 Amazon ECR 리포지토리

가장 중요한 조사결과가 있는 Elastic Container Registry(ECR) 리포지토리 섹션에는 사용 환경에서 가장 중요한 컨테이너 이미지 조사 결과가 있는 상위 5개 Amazon ECR 리포지토리가 표시됩니다. 이 보기에는 리포지토리 이름, AWS 계정 ID, 리포지토리 생성 날짜, 중요 취약성 수, 총 취약성 수가 표시됩니다. 이 보기는 가장 위험할 수 있는 리포지토리를 식별하는 데 도움이 됩니다.

리포지토리 이름을 선택하면 영향을 받는 리포지토리에 대한 자세한 정보를 볼 수 있습니다.

가장 중요한 조사 결과가 있는 컨테이너 이미지

가장 중요한 조사 결과가 있는 컨테이너 이미지 섹션에는 사용 환경에서 가장 중요한 조사 결과가 있는 상위 5개 컨테이너 이미지가 표시됩니다. 이 보기에는 이미지 태그 데이터, 리포지토리 이름, 이미지 다이제스트, AWS 계정 ID, 중요 취약성 수, 총 취약성 수가 표시됩니다. 이 보기는 애플리케이션 소유자가 재구축하여 재실행해야 하는 컨테이너 이미지를 파악하는 데 도움이 됩니다.

컨테이너 이미지를 선택하면 영향을 받는 컨테이너 이미지에 대한 자세한 정보를 볼 수 있습니다.

가장 중요한 조사 결과가 있는 인스턴스

가장 중요한 조사 결과가 있는 인스턴스 섹션에는 가장 중요한 조사 결과가 있는 상위 5개 Amazon EC2 인스턴스가 표시됩니다. 이 보기에는 인스턴스 식별자, AWS 계정 ID, Amazon Machine

Image(AMI) 식별자, 중요 취약성 수, 총 취약성 수가 표시됩니다. 이 보기는 인프라 소유자가 패치 적용이 필요한 인스턴스를 파악하는 데 도움이 됩니다.

인스턴스 ID를 선택하면 영향을 받는 Amazon EC2 인스턴스에 대한 자세한 정보를 볼 수 있습니다.

가장 중요한 조사 결과가 있는 Amazon Machine Image(AMI)

가장 중요한 조사 결과가 있는 Amazon Machine Image(AMI) 섹션에는 사용 환경에서 가장 중요한 조사 결과가 있는 상위 5개 AMI가 표시됩니다. 이 보기에는 AMI 식별자, AWS 계정 ID, 환경에서 실행 중인 영향을 받는 EC2 인스턴스 수, AMI 생성 날짜, AMI의 운영 체제 플랫폼, 중요 취약성 수, 총 취약성 수가 표시됩니다. 이 보기는 인프라 소유자가 재구축이 필요한 AMI를 식별하는 데 도움이 됩니다.

영향을 받는 인스턴스를 선택하면 영향을 받는 AMI에서 실행된 인스턴스에 대한 자세한 정보를 볼 수 있습니다.

가장 중요한 조사 결과가 있는 AWS Lambda 함수

가장 중요한 조사 결과가 있는 AWS Lambda 함수 섹션에는 사용 환경에서 가장 중요한 조사 결과가 있는 상위 5개 Lambda 함수가 표시됩니다. 이 보기에는 Lambda 함수 이름, AWS 계정 ID, 런타임 환경, 중요 취약성 수, 취약성이 높은 항목 수, 총 취약성 수가 표시됩니다. 이 보기는 인프라 소유자가 해결이 필요한 Lambda 함수를 파악하는 데 도움이 됩니다.

함수 이름을 선택하면 영향을 받는 AWS Lambda 함수에 대한 자세한 정보를 볼 수 있습니다.

가장 중요한 조사 결과를 사용한 Amazon Inspector 코드 스캔

가장 중요한 코드 취약성이 있는 프로젝트 섹션에는 중요한 조사 결과가 있는 상위 5개 프로젝트가 표시됩니다. 프로젝트를 선택하여 조사 결과에 대한 세부 정보를 볼 수 있습니다. 프로젝트를 선택하면 조사 결과가 있는 리포지토리로 이동합니다. 조사 결과 탭에는 조사 결과의 이름과 심각도 등급이 표시됩니다. 조사 결과를 생성하는 데 사용된 분석 유형을 보여줍니다. 또한 조사 결과의 수명과 상태를 보여줍니다.

Amazon Inspector 취약성 데이터베이스 검색

Amazon Inspector 취약성 데이터베이스에서 일반적인 취약성 및 노출(CVE)을 검색할 수 있습니다. Amazon Inspector는 취약성 데이터베이스의 정보를 사용하여 CVE ID와 관련된 세부 정보를 생성합니다. 이러한 세부 정보는 CVE 세부 정보 화면에서 볼 수 있습니다. Amazon Inspector는 취약성 데이터베이스에서 소프트웨어 취약성에 대한 조사 결과를 추적하고 생성합니다. Amazon Inspector는 CVE 세부 정보 화면의 탐지 플랫폼 섹션에 나열된 플랫폼의 CVE만 지원합니다. 이 단원에서는 CVE ID를 사용하여 Amazon Inspector 취약성 데이터베이스를 검색하는 방법에 대해 설명합니다.

> **Note**
>
> 현재 CVE 검색에서는 Microsoft Windows가 지원되지 않습니다.

취약성 데이터베이스 검색

이 단원에서는 콘솔과 Amazon Inspector API를 사용하여 취약성 데이터베이스를 검색하는 방법에 대해 설명합니다.

> **Note**
>
> 취약성 데이터베이스를 검색 AWS 리전 하려면 먼저 현재에서 Amazon Inspector를 활성화해야 합니다.

Console

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 취약성 데이터베이스 검색을 선택합니다.
3. 검색 창에 CVE ID를 입력하고 검색을 선택합니다.

API

Amazon Inspector SearchVulnerabilities API를 실행하고 단일 CVE ID를 `CVE-<year>-<ID>` 형식의 `filterCriteria`로 제공합니다.

CVE 세부 정보 이해

이 단원에서는 CVE 세부 정보 페이지를 해석하는 방법에 대해 설명합니다.

CVE 세부 정보

CVE 세부 정보 섹션은 다음 정보로 구성되어 있습니다.

- CVE 설명 및 ID
- CVE 심각도
- 공통 취약성 점수 시스템(CVSS) 및 공격 예측 점수 시스템(EPSS) 점수
- 탐지 플랫폼

> **Note**
>
> 이 필드가 비어 있으면 Amazon Inspector에서 CVE ID에 대한 탐지를 지원하지 않는 것입니다.

- CWE(Common Weakness Enumeration)
- 공급업체 생성 및 업데이트 날짜

취약성 인텔리전스

취약성 인텔리전스 섹션에서는 공격 대상 및 마지막으로 알려진 공개 공격 날짜와 같은 위협 인텔리전스 데이터를 제공합니다.

또한 사이버 보안 및 인프라 보안국(CISA)의 데이터도 제공합니다. 여기에는 해결 작업, 알려진 악용 취약성 카탈로그에 CVE가 추가된 날짜, 연방 기관이 CVE를 수정할 것으로 CISA에서 예상하는 날짜가 포함됩니다.

참조

참조 섹션에서는 CVE에 대한 자세한 내용을 볼 수 있는 리소스 링크를 제공합니다.

Amazon Inspector를 사용하여 SBOM 내보내기

소프트웨어 자재 명세서(SBOM)는 코드베이스에 있는 모든 오픈 소스 및 타사 소프트웨어 구성 요소의 중첩된 인벤토리입니다. Amazon Inspector는 사용 환경의 개별 리소스에 대한 SBOM을 제공합니다. Amazon Inspector 콘솔 또는 Amazon Inspector API를 사용하여 리소스에 대한 SBOM을 생성할 수 있습니다. Amazon Inspector에서 지원하고 모니터링하는 모든 리소스에 대한 SBOM을 내보낼 수 있습니다. 내보낸 SBOM은 소프트웨어 공급에 대한 정보를 제공합니다. AWS 환경의 적용 범위를 평가하여 리소스 상태를 검토할 수 있습니다. 이 단원에서는 SBOM을 구성하고 내보내는 방법에 대해 설명합니다.

일부 소프트웨어 구성 요소 및 패키지 관리자는 종속성에 고정 버전 대신 버전 범위 또는 동적 참조를 사용합니다. 이 방법은 Amazon Inspector가 해시 또는 jar 파일을 식별하지만 취약성 감지를 위해 특정 이름 및 버전에 매핑할 수 없는 해결되지 않은 해시를 생성합니다. 이제 Amazon Inspector는 해결되지 않은 이러한 해시를 소프트웨어 재료표(SBOM) 내보내기에 포함합니다. 이러한 패키지는 취약성을 스캔할 수 없지만 내보낸 구성 요소 목록에서 해당 해시 값을 사용할 수 있습니다.

> **Note**
>
> 현재 Amazon Inspector는 Windows Amazon EC2 인스턴스에 대한 SBOM 내보내기를 지원하지 않습니다.

Amazon Inspector 형식

Amazon Inspector는 CycloneDX 1.4 및 SPDX 2.3 호환 형식으로 SBOM 내보내기를 지원합니다. Amazon Inspector는 선택한 Amazon S3 버킷에 SBOM을 `JSON` 파일로 내보냅니다.

> **Note**
>
> Amazon Inspector에서 내보내는 SPDX 형식은 SPDX 2.3을 사용하는 시스템과 호환되지만, Creative Commons Zero(CC0) 필드가 포함되어 있지 않습니다. 이 필드가 포함되어 있으면 사용자가 자료를 재배포하거나 편집할 수 있기 때문입니다.

Amazon Inspector의 CyclonedX 1.4 SBOM 형식 예제

```
{
```

```
  "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"
        }
      ]
    }
  ]
}
```

Amazon Inspector의 SPDX 2.3 SBOM 형식 예제


```
{
 "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"
}
```


SBOM용 필터

SBOM을 내보낼 때 필터를 포함시켜 리소스의 특정 하위 집합에 대한 보고서를 생성할 수 있습니다. 필터를 제공하지 않으면 지원되는 모든 활성 리소스의 SBOM을 내보냅니다. 또한 위임 관리자인 경우 여기에 모든 멤버를 위한 리소스도 포함됩니다. 다음과 같은 필터를 사용할 수 있습니다.

- AccountID - 이 필터는 특정 계정 ID와 연결된 리소스의 SBOM을 내보내는 데 사용할 수 있습니다.
- EC2 인스턴스 태그 - 이 필터는 특정 태그가 있는 EC2 인스턴스의 SBOM을 내보내는 데 사용할 수 있습니다.

- 함수 이름 - 이 필터는 특정 Lambda 함수의 SBOM을 내보내는 데 사용할 수 있습니다.
- 이미지 태그 - 이 필터는 특정 태그가 있는 컨테이너 이미지의 SBOM을 내보내는 데 사용할 수 있습니다.
- Lambda 함수 태그 - 이 필터는 특정 태그가 있는 Lambda 함수의 SBOM을 내보내는 데 사용할 수 있습니다.
- 리소스 유형 - 이 필터는 리소스 유형(EC2/ECR/Lambda)을 필터링하는 데 사용할 수 있습니다.
- 리소스 ID - 이 필터는 특정 리소스의 SBOM을 내보내는 데 사용할 수 있습니다.
- 리포지토리 이름 - 이 필터는 특정 리포지토리에 있는 컨테이너 이미지의 SBOM을 생성하는 데 사용할 수 있습니다.

SBOM 구성 및 내보내기

SBOM을 내보내려면 먼저 Amazon Inspector에서 사용할 수 있는 Amazon S3 버킷과 AWS KMS 키를 구성해야 합니다. 필터를 사용하여 리소스의 특정 하위 집합에 대한 SBOM을 내보낼 수 있습니다. AWS Organization 내 여러 계정의 SBOM을 내보내려면 Amazon Inspector 위임 관리자로 로그인한 상태에서 다음 단계를 수행합니다.

사전 조건

- Amazon Inspector에서 적극적으로 모니터링하고 있는 지원 리소스
- Amazon Inspector에서 객체를 추가할 수 있도록 허용하는 정책으로 구성된 Amazon S3 버킷. 정책 구성에 대한 자세한 내용은 내보내기 권한 구성을 참조하세요.
- Amazon Inspector에서 보고서를 암호화하는 데 사용할 수 있도록 허용하는 정책으로 구성된 AWS KMS 키. 정책 구성에 대한 자세한 내용은 Configure an AWS KMS key for export를 참조하세요.

> **Note**
>
> 이전에 조사 결과 내보내기용 Amazon S3 버킷과 AWS KMS 키를 구성한 경우 동일한 버킷과 키를 SBOM 내보내기에 사용할 수 있습니다.

원하는 액세스 방법을 선택하여 SBOM을 내보냅니다.

Console

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 페이지 오른쪽 상단에 있는 AWS 리전 선택기를 사용하여 SBOM을 내보낼 리소스가 있는 리전을 선택합니다.
3. 탐색 창에서 SBOM 내보내기를 선택합니다.
4. (선택 사항) SBOM 내보내기 페이지에서 필터 추가 메뉴를 사용하여 보고서를 생성할 리소스의 하위 집합을 선택합니다. 필터를 제공하지 않으면 Amazon Inspector에서 모든 활성 리소스에 대한 보고서를 내보냅니다. 위임 관리자인 경우 여기에 조직의 모든 활성 리소스가 포함됩니다.
5. 내보내기 설정에서 SBOM에 사용할 형식을 선택합니다.
6. Amazon S3 URI를 입력하거나 Amazon S3 찾아보기를 선택하여 SBOM을 저장할 Amazon S3 위치를 선택합니다.
7. Amazon Inspector에서 보고서를 암호화하는 데 사용하도록 구성된 AWS KMS 키를 입력합니다.

API

- 리소스의 SBOM을 프로그래밍 방식으로 내보내려면 Amazon Inspector API의 CreatesBomExport 작업을 사용합니다.

  요청에서 `reportFormat` 파라미터를 사용하여 SBOM 출력 형식을 지정하고 `CYCLONEDX_1_4` 또는 `SPDX_2_3`을 선택합니다. `s3Destination` 파라미터는 필수이며, Amazon Inspector에서 쓰기를 허용하는 정책으로 구성된 S3 버킷을 지정해야 합니다. 선택적으로 `resourceFilterCriteria` 파라미터를 사용하여 보고서의 범위를 특정 리소스로 제한할 수 있습니다.

AWS CLI

- AWS Command Line Interface를 사용하여 리소스의 sBOM을 내보내려면 다음 명령을 실행합니다.

  ```
  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
  ```

요청에서 *FORMAT*을 CYCLONEDX_1_4 또는 SPDX_2_3 중에서 원하는 형식으로 바꾸세요. 그런 다음 s3 대상의 *user input placeholders*를 내보낼 대상 S3 버킷의 이름, S3의 출력에 사용할 접두사, 보고서를 암호화하는 데 사용하는 KMS 키의 ARN으로 바꾸세요.

Amazon Inspector 이벤트에 대한 Amazon EventBridge 이벤트 스키마

Amazon EventBridge는 애플리케이션 및 기타 AWS 서비스 서비스의 실시간 데이터 스트림을 AWS Lambda 함수, Amazon Simple Notification Service 주제, Amazon Kinesis Data Streams의 데이터 스트림과 같은 대상에 전달합니다. 다른 애플리케이션, 서비스 및 시스템과의 통합을 지원하기 위해 Amazon Inspector에서는 조사 결과를 자동으로 Amazon EventBridge에 이벤트로 게시합니다. Amazon Inspector를 사용하여 조사 결과, 적용 범위 및 스캔에 대한 이벤트를 게시할 수 있습니다. 이 단원에서는 EventBridge 이벤트에 대한 예제 스키마를 제공합니다.

주제

- Amazon Inspector의 Amazon EventBridge 기본 스키마
- Amazon Inspector 조사 결과 이벤트 스키마 예제
- Amazon Inspector 최초 스캔 완료 이벤트 스키마 예제
- Amazon Inspector 적용 범위 이벤트 스키마 예제
- Amazon Inspector 스키마 자동 활성화 예제

Amazon Inspector의 Amazon EventBridge 기본 스키마

다음은 Amazon Inspector의 EventBridge 이벤트에 대한 기본 스키마 예제입니다. 이벤트 세부 정보는 이벤트 유형에 따라 다릅니다.

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

```
}
```

Amazon Inspector 조사 결과 이벤트 스키마 예제

다음은 Amazon Inspector 조사 결과에 대한 EventBridge 이벤트 스키마의 예입니다. 조사 결과 이벤트는 Amazon Inspector가 리소스 중 하나에서 소프트웨어 취약성 또는 네트워크 문제를 식별할 경우에 생성됩니다. 이 유형의 이벤트에 대한 대응으로 알림을 생성하는 방법에 대한 지침은 Amazon EventBridge를 사용하여 Amazon Inspector 조사 결과에 대한 사용자 지정 응답 생성을 참조하세요.

다음은 조사 결과 이벤트를 식별하는 필드입니다.

- `detail-type`를 (으)로 설정합니다.`Inspector2 Finding`
- `detail`은 조사 결과를 설명합니다.
- `detail.resources.tags`는 키-값 데이터가 저장되는 위치입니다.

탭을 필터링하여 다양한 리소스에 대한 조사 결과 이벤트 스키마와 조사 결과 유형을 확인할 수 있습니다.

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": [
                            "1.23.456.789",
                            "123.45.67.890"
                        ],
                        "ipV6Addresses": [],
                        "launchedAt": "Wed Sep 04 16:57:40.000 UTC 2024",
                        "platform": "UBUNTU_22_04",
                        "subnetId": "subnet-12345678",
                        "type": "t2.small",
                        "vpcId": "vpc-12345678"
                    }
                },
                "id": "i-12345678901234567",
```

```
                "partition": "aws",
                "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": ["1.23.456.789", "123.45.67.890"],
                    "ipV6Addresses": [],
                    "launchedAt": "Wed Sep 04 17:41:24.000 UTC 2024",
                    "platform": "UBUNTU_22_04",
                    "subnetId": "subnet-12345678",
                    "type": "t2.small",
                    "vpcId": "vpc-12345678"
                }
            },
            "id": "i-12345678901234567",
            "partition": "aws",
            "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"
                    }
                },
                "id": "arn:aws:ecr:eu-central-1:123456789012:repository/inspector2/
sha256:84f507df33c6864d49c296fb734192696e4cb6f78166ac51ac8b9b118181085d",
                "partition": "aws",
                "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": "$LATEST"
                    }
                },
                "id": "arn:aws:lambda:eu-
central-1:123456789012:function:VulnerableFunction:$LATEST",
                "partition": "aws",
                "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": "$LATEST"
                    }
                },
                "id": "arn:aws:lambda:eu-
central-1:123456789012:function:VulnerableFunction:$LATEST",
                "partition": "aws",
                "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"
    }
}
```


> **Note**
>
> detail 값은 단일 조사 결과에 대한 JSON 세부 정보를 객체로 반환합니다. 배열 내의 여러 조사 결과를 지원하는 전체 결과 응답 구문은 반환하지 않습니다.

Amazon Inspector 최초 스캔 완료 이벤트 스키마 예제

다음은 최초 스캔 완료에 대한 Amazon Inspector 이벤트의 EventBridge 이벤트 스키마 예제입니다. 이 이벤트는 Amazon Inspector에서 리소스 중 하나에 대한 최초 스캔을 완료할 경우에 생성됩니다.

다음은 최초 스캔 완료 이벤트를 식별하는 필드입니다.

- `detail-type` 필드가 `Inspector2 Scan`으로 설정되어 있습니다.

- `detail` 객체에는 해당 심각도 범주(예: `CRITICAL`, `HIGH`, `MEDIUM`)에 있는 조사 결과 수를 자세히 설명하는 `finding-severity-counts` 객체가 포함되어 있습니다.

아래 옵션 중 하나를 선택하면 여러 최초 스캔 이벤트 스키마를 리소스 유형별로 확인할 수 있습니다.

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"
  }
}
```

Amazon Inspector 적용 범위 이벤트 스키마 예제

다음은 적용 범위에 대한 Amazon Inspector 이벤트의 EventBridge 이벤트 스키마 예제입니다. 이 이벤트는 리소스에 대한 Amazon Inspector 스캔 적용 범위가 변경될 때 생성됩니다. 다음은 적용 범위 이벤트를 식별하는 필드입니다.

- `detail-type` 필드가 `Inspector2 Coverage`으로 설정되어 있습니다.
- `detail` 객체에는 리소스의 새 스캔 상태를 나타내는 `scanStatus` 객체가 포함되어 있습니다.

```
{
    "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",
        "eventTimestamp": "2023-01-20T22:51:35.665501Z",
        "version": "1.0"
```

```
    }
}
```

Amazon Inspector 스키마 자동 활성화 예제

Amazon Inspector가 조직의 멤버 수를 지원할 수 없는 경우 자동 활성화 이벤트가 위임된 관리자에게 전송됩니다. 다음 필드는 자동 활성화 이벤트를 식별합니다.

- `detail-type` 필드가 `Inspector2 AutoEnable`으로 설정되어 있습니다.
- `detail` 객체는 자동 활성화 이벤트가 실패한 이유를 설명합니다.

```
{
    "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"
        }
}
```

Linux 및 Windows용 Amazon Inspector SSM 플러그인

이 주제에서는 Linux 및 Windows 인스턴스에 대한 Amazon Inspector SSM 플러그인을 설명합니다.

Linux용 Amazon Inspector SSM 플러그인

Amazon Inspector는 Amazon Inspector SSM 플러그인을 사용하여 Linux 인스턴스에 대한 심층 검사 스캔을 수행합니다. Amazon Inspector SSM 플러그인은 Linux 인스턴스의 `/opt/aws/inspector/bin` 디렉터리에 자동으로 설치됩니다. 실행 파일의 이름은 `inspectorssmplugin`입니다.

Amazon Inspector는 Systems Manager Distributor를 사용하여 인스턴스에 플러그인을 배포합니다. 심층 검사 스캔을 수행하려면 Systems Manager Distributor 및 Amazon Inspector에서 Amazon EC2 인스턴스 운영 체제를 지원해야 합니다. Systems Manager Distributor에서 지원하는 운영 체제에 대한 자세한 내용은 AWS Systems Manager 사용 설명서에서 지원되는 패키지 플랫폼 및 아키텍처를 참조하세요.

Amazon Inspector SSM 플러그인에서 수집한 심층 검사용 데이터를 관리하기 위해 Amazon Inspector는 파일 디렉터리를 생성합니다. 이러한 파일 디렉터리에는 `/opt/aws/inspector/var/input` 및 `/opt/aws/inspector/var/output`가 포함됩니다.

`/opt/aws/inspector/var/output`의 `packages.txt` 파일에는 심층 검사로 발견된 패키지의 전체 경로가 저장됩니다. Amazon Inspector가 인스턴스에서 동일한 패키지를 여러 번 탐지한 경우 `packages.txt` 파일에는 해당 패키지가 발견된 각 위치가 나열됩니다.

Amazon Inspector는 플러그인에 대한 로그를 `/var/log/amazon/inspector` 디렉터리에 저장합니다.

Amazon Inspector SSM 플러그인 제거

`inspectorssmplugin` 파일이 실수로 삭제된 경우 SSM 연결 `InspectorLinuxDistributor-do-not-delete`를 통해 다음 스캔 간격에 `inspectorssmplugin` 파일을 다시 설치하려고 시도합니다.

Amazon EC2 스캔을 비활성화하면 모든 Linux 호스트에서 플러그인이 자동으로 제거됩니다.

Windows용 Amazon Inspector SSM 플러그인

Amazon Inspector에서 Windows 인스턴스를 스캔하려면 Amazon Inspector SSM 플러그인이 필요합니다. Amazon Inspector SSM 플러그인은 Windows 인스턴스의 `C:\Program`

Files\Amazon\Inspector에 자동으로 설치되며 실행 가능한 바이너리 파일의 이름은 InspectorSsmPlugin.exe입니다.

Amazon Inspector SSM 플러그인에서 수집한 데이터를 저장하기 위해 다음과 같은 파일 위치가 생성됩니다.

- C:\ProgramData\Amazon\Inspector\Input
- C:\ProgramData\Amazon\Inspector\Output
- C:\ProgramData\Amazon\Inspector\Logs

> **Note**
>
> 기본적으로 Amazon Inspector SSM 플러그인은 일반 우선순위보다 낮은 우선 순위로 실행됩니다.

> **Note**
>
> Windows 인스턴스는 기본 호스트 관리 구성 설정으로 사용할 수 있습니다. 그러나 ssm:PutInventory 및 ssm:GetParameter 권한으로 구성된 역할을 생성하거나 사용해야 합니다.

Amazon Inspector SSM 플러그인 제거

InspectorSsmPlugin.exe 파일이 실수로 삭제된 경우 InspectorDistributor-do-not-delete 연결을 통해 다음 Windows 스캔 간격에 InspectorSsmPlugin.exe 파일을 다시 설치하려고 시도합니다. Amazon Inspector SSM 플러그인을 제거하려는 경우 AmazonInspector2-ConfigureInspectorSsmPlugin 문서에서 제거 작업을 사용하면 됩니다. 또한 Amazon EC2 스캔을 비활성화하면 Amazon Inspector SSM 플러그인이 모든 Windows 호스트에서 자동으로 제거됩니다.

> **Note**
>
> Amazon Inspector를 비활성화하기 전에 SSM 에이전트를 제거하면 Amazon Inspector SSM 플러그인은 Windows 호스트에 남아 있지만 Amazon Inspector SSM 플러그인으로 데이터를 보내지 않습니다. 자세한 내용은 Amazon Inspector 비활성화 섹션을 참조하세요.

Amazon Inspector SBOM 생성기

소프트웨어 자재 명세서(SBOM)는 소프트웨어를 빌드하는 데 필요한 구성 요소, 라이브러리 및 모듈을 정식으로 구조화한 목록입니다. Amazon Inspector SBOM 생성기(Sbomgen)는 아카이브, 컨테이너 이미지, 디렉토리, 로컬 시스템, 컴파일된 Go 및 Rust 바이너리에 대한 SBOM을 생성하는 도구입니다. Sbomgen은 설치된 패키지에 대한 정보가 포함된 파일을 스캔합니다. Sbomgen에서 관련 파일을 찾으면 패키지 이름, 버전 및 기타 메타데이터를 추출합니다. 그런 다음 Sbomgen은 패키지 메타데이터를 CycloneDX SBOM으로 변환합니다. 사용자는 Sbomgen을 사용하여 CycloneDX SBOM을 파일 또는 STDOUT으로 생성하고 취약성 탐지를 위해 SBOM을 Amazon Inspector로 전송할 수 있습니다. 또한 배포 파이프라인의 일부로 컨테이너 이미지를 자동으로 스캔하는 CI/CD 통합의 일부로 Sbomgen을 사용할 수도 있습니다.

지원되는 패키지 유형

Sbomgen은 다음 패키지 유형에 대한 인벤토리를 수집합니다.

- Alpine APK
- Debian/Ubuntu DPKG
- Red Hat RPM
- C#
- Go
- Java
- Node.js
- PHP
- Python
- Ruby
- Rust

지원되는 컨테이너 이미지 구성 검사

Sbomgen은 독립 실행형 Dockerfile과 기존 이미지의 빌드 이력을 스캔하여 보안 문제를 점검할 수 있습니다. 자세한 내용은 Amazon Inspector Dockerfile 검사를 참조하세요.

Sbomgen 설치하기

Sbomgen은 Linux 운영 체제에서만 사용할 수 있습니다.

Sbomgen이 로컬에 캐시된 이미지를 분석하려면 Docker가 설치되어 있어야 합니다. `.tar` 파일로 내보낸 이미지나 원격 컨테이너 레지스트리에서 호스팅된 이미지를 분석하는 데는 Docker가 필요하지 않습니다.

Amazon Inspector는 최소한 다음 하드웨어 사양을 갖춘 시스템에서 Sbomgen을 실행할 것을 권장합니다.

- 4x 코어 CPU
- 8GB RAM

Sbomgen을(를) 설치하려면

1. 아키텍처에 맞는 올바른 URL에서 최신 Sbomgen zip 파일을 다운로드합니다.

   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

   또는 Amazon Inspector SBOM 생성기 zip 파일의 이전 버전을 다운로드할 수 있습니다.
2. 다음 명령을 사용하여 다운로드를 압축 해제합니다.

   ```
   unzip inspector-sbomgen.zip
   ```
3. 추출된 디렉터리에서 다음 파일을 확인합니다.

   - `inspector-sbomgen` - SBOM을 생성하기 위해 실행하는 도구입니다.
   - `README.txt` - Sbomgen 사용 설명서입니다.
   - `LICENSE.txt` - 이 파일에는 Sbomgen에 대한 소프트웨어 라이선스가 들어 있습니다.
   - `licenses` - 이 폴더에는 Sbomgen에서 사용하는 서드 파티 패키지의 라이선스 정보가 들어 있습니다.
   - `checksums.txt` - 이 파일은 Sbomgen 도구의 해시를 제공합니다.
   - `sbom.json` - Sbomgen 도구용 CycloneDX SBOM입니다.

- `WhatsNew.txt` - 이 파일에는 요약된 변경 로그가 포함되어 있으므로 Sbomgen 버전 간의 주요 변경 사항과 개선 사항을 빠르게 확인할 수 있습니다.

4. (선택 사항) 다음 명령을 사용하여 도구의 신뢰성과 무결성을 확인합니다.

   ```
   sha256sum < inspector-sbomgen
   ```

   - 결과를 `checksums.txt` 파일 콘텐츠와 비교하세요.

5. 다음 명령을 사용하여 도구에 실행 권한을 부여합니다.

   ```
   chmod +x inspector-sbomgen
   ```

6. 다음 명령을 실행하여 Sbomgen이 성공적으로 설치되었는지 확인합니다.

   ```
   ./inspector-sbomgen --version
   ```

   다음과 유사한 출력 화면이 표시되어야 합니다.

   ```
   Version: 1.X.X
   ```

Sbomgen 사용하기

이 단원에서는 Sbomgen을 사용할 수 있는 다양한 방법에 대해 설명합니다. 기본 제공 예제를 통해 Sbomgen 사용 방법에 대해 자세히 알아볼 수 있습니다. 이러한 예제를 보려면 `list-examples` 명령을 실행합니다.

```
./inspector-sbomgen list-examples
```

컨테이너 이미지에 대한 SBOM 생성 및 결과 출력

Sbomgen을 사용하여 컨테이너 이미지에 대한 SBOM을 생성하고 결과를 파일로 출력할 수 있습니다. 이 기능은 `container` 하위 명령을 사용하여 활성화할 수 있습니다.

명령 예제:

다음 스니펫에서 *`image:tag`*는 이미지 ID로 바꾸고, *`output_path.json`*은 저장하려는 출력 경로로 바꿀 수 있습니다.

```
# generate SBOM for container image
./inspector-sbomgen container --image image:tag -o output_path.json
```

> **Note**
>
> 스캔 시간과 성능은 이미지 크기와 계층 수에 따라 달라집니다. 이미지가 작을수록 Sbomgen 성능이 향상될 뿐만 아니라 잠재적인 공격 표면도 줄어듭니다. 이미지가 작을수록 이미지 빌드, 다운로드, 업로드 시간도 단축됩니다.

와 Sbomgen 함께 ScanSbom를 사용하는 경우 Amazon Inspector 스캔 API는 5,000개 이상의 패키지가 포함된 SBOMs 처리하지 않습니다. 이 시나리오에서 Amazon Inspector 스캔 API는 HTTP 400 응답을 반환합니다.

이미지에 대용량 미디어 파일 또는 디렉터리가 포함된 경우 `--skip-files` 인수를 사용하여 Sbomgen에서 이를 제외하는 것이 좋습니다.

예: 일반적인 오류 사례

컨테이너 이미지 스캔은 다음 오류로 인해 실패할 수 있습니다.

- `InvalidImageFormat` - 잘못된 형식의 컨테이너 이미지를 손상된 TAR 헤더, 매니페스트 파일 또는 구성 파일로 스캔할 때 발생합니다.
- `ImageValidationFailure` - 일치하지 않는 Content-Length 헤더, 잘못된 매니페스트 다이제스트 또는 실패한 SHA256 체크섬 확인과 같은 컨테이너 이미지 구성 요소에 대해 체크섬 또는 콘텐츠 길이 검증이 실패할 때 발생합니다.
- `ErrUnsupportedMediaType` - 이미지 구성 요소에 지원되지 않는 미디어 유형이 포함된 경우 발생합니다. 지원되는 미디어 유형에 대한 자세한 내용은 지원되는 운영 체제 및 미디어 유형을 참조하세요.

Amazon Inspector는 `application/vnd.docker.distribution.manifest.list.v2+json` 미디어 유형을 지원하지 않습니다. 그러나 Amazon Inspector는 매니페스트 목록을 지원합니다. 매니페스트 목록을 사용하는 이미지를 스캔할 때 `--platform` 인수와 함께 사용할 플랫폼을 명시적으로 지정할 수 있습니다. `--platform` 인수를 지정하지 않으면 Amazon Inspector SBOM 생성기가 실행 중인 플랫폼을 기반으로 매니페스트를 자동으로 선택합니다.

디렉터리 및 아카이브에서 SBOM 생성

Sbomgen을 사용하여 디렉터리 및 아카이브에서 SBOM을 생성할 수 있습니다. 이 기능은 `directory` 또는 `archive` 하위 명령을 사용하여 활성화할 수 있습니다. Amazon Inspector는 다운로드한 git 리포지토리와 같은 프로젝트 폴더에서 SBOM을 생성하려는 경우에 이 특성을 사용할 것을 권장합니다.

명령 예제 1

다음 스니펫은 디렉터리 파일에서 SBOM을 생성하는 하위 명령을 보여줍니다.

```
# generate SBOM from directory
./inspector-sbomgen directory --path /path/to/dir -o /tmp/sbom.json
```

명령 예제 2

다음 스니펫은 아카이브 파일에서 SBOM을 생성하는 하위 명령을 보여줍니다. `.zip`, `.tar`, `.tar.gz` 아카이브 형식만 지원됩니다.

```
# generate SBOM from archive file (tar, tar.gz, and zip formats only)
./inspector-sbomgen archive --path testData.zip -o /tmp/sbom.json
```

Go 또는 Rust 컴파일된 바이너리에서 SBOM 생성

Sbomgen을 사용하여 컴파일된 Go 및 Rust 바이너리에서 SBOM을 생성할 수 있습니다. 이 기능은 `binary` 하위 명령을 통해 활성화할 수 있습니다.

```
./inspector-sbomgen binary --path /path/to/your/binary
```

탑재된 볼륨에서 SBOM 생성

Amazon Inspector SBOM 생성기를 사용하여 탑재된 볼륨에서 SBOM을 생성할 수 있습니다. 이 기능은 `volume` 하위 명령을 사용하여 활성화할 수 있습니다. 시스템에 탑재된 Amazon EBS 볼륨과 같은 스토리지 볼륨을 분석하려는 경우이 특성을 사용하는 것이 좋습니다. 디렉터리 하위 명령과 달리 탑재 볼륨 스캔은 OS 패키지 및 OS 정보를 감지합니다.

Amazon EBS 볼륨을 Amazon Inspector SBOM 생성기가 설치된 Amazon EC2 인스턴스에 연결하고 해당 인스턴스에 탑재하여 스캔할 수 있습니다. 현재 다른 Amazon EC2 인스턴스에서 사용 중인 Amazon EBS 볼륨의 경우 볼륨의 Amazon EBS 스냅샷을 생성한 다음 스캔 목적으로 해당 스냅샷에서 새 Amazon EBS 볼륨을 생성할 수 있습니다. Amazon EBS에 대한 자세한 내용은 Amazon Elastic Block Store 사용 설명서에서 Amazon EBS란? 섹션을 참조하세요.

명령 예제:

다음 코드 조각은 탑재된 볼륨에서 SBOM을 생성하는 하위 명령을 보여줍니다. `--path` 인수는 볼륨이 탑재되는 루트 디렉터리를 지정해야 합니다.

```
# generate SBOM from mounted volume
./inspector-sbomgen volume --path /mount/point/of/volume/root
```

명령 예제:

다음 코드 조각은 `--exclude-suffix` 인수가 있는 특정 파일 경로를 제외하면서 탑재된 볼륨에서 SBOM을 생성하는 하위 명령을 보여줍니다. `--exclude-suffix` 인수는 볼륨에 대량 파일(예: 로그 파일 또는 미디어 파일)이 포함된 경우 특히 유용합니다. 경로가 지정된 접미사로 끝나는 파일 및 디렉터리는 스캔에서 제외되므로 스캔 시간과 메모리 사용량을 줄일 수 있습니다.

```
# generate SBOM from mounted volume with exclusions
./inspector-sbomgen volume --path /mount/point/of/volume/root \
--exclude-suffix .log \
--exclude-suffix cache
```

대상 볼륨의 모든 파일 경로는 원래 경로로 정규화됩니다. 예를 들어, 파일이 포함된 `/mnt/volume`에 탑재된 볼륨을 `/mnt/volume/var/lib/rpm/rpmdb.sqlite`에서 스캔할 때 경로는 생성된 SBOM에서 `/var/lib/rpm/rpmdb.sqlite`로 정규화됩니다.

취약성 식별을 위해 Amazon Inspector로 SBOM 전송

SBOM을 생성하는 것 외에도 Amazon Inspector 스캔 API에서 한 번의 명령으로 스캔을 위해 SBOM을 전송할 수 있습니다. Amazon Inspector는 SBOM의 내용을 평가하여 취약성이 있는지 확인한 후 조사 결과를 Sbomgen에 반환합니다. 사용자의 입력에 따라 조사 결과를 표시하거나 파일에 기록할 수 있습니다.

> **Note**
>
> 이 기능을 사용하려면에 `InspectorScan-ScanSbom` 대한 읽기 권한이 AWS 계정 있는 활성이 있어야 합니다.

이 기능을 활성화하려면 `--scan-sbom` 인수를 Sbomgen CLI에 전달합니다. `--scan-sbom` 인수를 Sbomgen 하위 명령 `archive`, `binary`, `container`, `directory`, `localhost` 중 하나에 전달할 수도 있습니다.

> **Note**
>
> Amazon Inspector 스캔 API는 5,000개 이상의 패키지가 있는 SBOMs 처리하지 않습니다. 이 시나리오에서 Amazon Inspector 스캔 API는 HTTP 400 응답을 반환합니다.

다음 AWS CLI 인수를 사용하여 AWS 프로필 또는 IAM 역할을 통해 Amazon Inspector에 인증할 수 있습니다.

```
--aws-profile profile
--aws-region region
--aws-iam-role-arn role_arn
```

다음 환경 변수를 Sbomgen에 제공하여 Amazon Inspector에 인증할 수도 있습니다.

```
AWS_ACCESS_KEY_ID=$access_key \
AWS_SECRET_ACCESS_KEY=$secret_key \
AWS_DEFAULT_REGION=$region \
./inspector-sbomgen arguments
```

응답 형식을 지정하려면 `--scan-sbom-output-format cyclonedx` 인수 또는 `--scan-sbom-output-format inspector` 인수를 사용합니다.

명령 예제 1

이 명령은 최신 Alpine Linux 릴리스에 대한 SBOM을 생성하고, SBOM을 스캔하고, 취약성 결과를 JSON 파일에 기록합니다.

```
./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
```

명령 예제 2

이 명령은 AWS 자격 증명을 환경 변수로 사용하여 Amazon Inspector에 인증합니다.

```
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
```

명령 예제 3

이 명령은 IAM 역할에 대한 ARN을 사용하여 Amazon Inspector에 인증합니다.

```
./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
```

추가 스캐너를 사용하여 탐지 기능 향상

Amazon Inspector SBOM 생성기는 사용 중인 명령을 기반으로 사전 정의된 스캐너를 적용합니다.

기본 스캐너 그룹

각 Amazon Inspector SBOM 생성기 하위 명령은 다음과 같은 기본 스캐너 그룹을 자동으로 적용합니다.

- `directory` 하위 명령의 경우: 바이너리, 프로그래밍 언어 패키지, dockerfile 스캐너 그룹
- `localhost` 하위 명령의 경우: os, 프로그래밍 언어 패키지, 추가 에코시스템 스캐너 그룹
- `container` 하위 명령의 경우: os, 프로그래밍 언어 패키지, 추가 에코시스템, dockerfile, 바이너리 스캐너 그룹

특수 스캐너

기본 스캐너 그룹 이외의 스캐너를 포함하려면 `--additional-scanners` 옵션과 추가할 스캐너 이름을 차례로 사용합니다. 다음은 이를 수행하는 방법을 보여주는 예제입니다.

```
# Add WordPress installation scanner to directory scan
./inspector-sbomgen directory --path /path/to/directory/ --additional-scanners
 wordpress-installation -o output.json
```

다음은 쉼표로 구분된 목록으로 여러 스캐너를 추가하는 방법을 보여주는 예제 명령입니다.

```
./inspector-sbomgen container --image image:tag --additional-scanners scanner1,scanner2
 -o output.json
```

스캔할 최대 파일 크기를 조정하여 컨테이너 스캔 최적화

컨테이너 이미지를 분석하고 처리할 때 Sbomgen는 기본적으로 200MB 이하의 파일을 스캔합니다. 200MB보다 큰 파일에는 패키지 메타데이터가 거의 포함되지 않습니다. 200MB를 초과하는 Go 또는 Rust 바이너리를 인벤토리할 때 누락이 발생할 수 있습니다. 크기 제한을 조정하려면 `--max-file-size` 인수를 사용합니다. 이렇게 하면 대용량 파일을 포함하도록 제한을 늘리고 대용량 파일을 제외하여 리소스 사용량을 줄이기 위해 제한을 줄일 수 있습니다.

예제

다음 예제에서는 `--max-file-size` 인수를 사용하여 파일 크기를 늘리는 방법을 보여줍니다.

```
# 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
```

이 설정을 조정하면 디스크 사용량, 메모리 사용량 및 전체 스캔 기간을 제어하는 데 도움이 됩니다.

진행률 표시기 비활성화

Sbomgen은 CI/CD 환경에서 과도한 슬래시 문자를 초래할 수 있는 회전 진행률 표시기를 표시합니다.

```
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
```

`--disable-progress-bar` 인수를 사용하여 진행률 표시기를 비활성화할 수 있습니다.

```
./inspector-sbomgen container --image alpine:latest \
--outfile /tmp/sbom.json \
--disable-progress-bar
```

Sbomgen을 사용하여 프라이빗 레지스트리에 인증

프라이빗 레지스트리 인증 자격 증명을 제공하면 프라이빗 레지스트리에서 호스팅되는 컨테이너에서 SBOM을 생성할 수 있습니다. 이러한 자격 증명은 다음 방법을 통해 제공할 수 있습니다.

캐시된 자격 증명을 사용하여 인증(권장)

이 방법을 사용하려면 컨테이너 레지스트리에 인증해야 합니다. 예를 들어 Docker를 사용하는 경우 Docker loging 명령 `docker login`을 사용하여 컨테이너 레지스트리에 인증할 수 있습니다.

1. 컨테이너 레지스트리에 인증합니다. 예를 들어 Docker를 사용하는 경우 Docker `login` 명령을 사용하여 레지스트리에 인증할 수 있습니다.
2. 컨테이너 레지스트리에 인증한 후 레지스트리에 있는 컨테이너 이미지에 Sbomgen을 사용합니다. 다음 예를 사용하려면 *`image:tag`*를 스캔할 이미지의 이름으로 바꾸세요.

```
./inspector-sbomgen container --image image:tag
```

대화형 방법을 사용한 인증

이 방법의 경우 사용자 이름을 파라미터로 제공하면 Sbomgen에서 필요할 때 보안 암호 입력을 요구하는 메시지를 표시합니다.

다음 예를 사용하려면 *`image:tag`*는 스캔할 이미지의 이름으로 바꾸고, *`your_username`*은 해당 이미지에 액세스할 수 사용자 이름으로 바꿉니다.

```
./inspector-sbomgen container --image image:tag --username your_username
```

비대화형 방법을 사용한 인증

이 방법의 경우 암호 또는 레지스트리 토큰을 `.txt` 파일에 저장합니다.

> **Note**
>
> 현재 사용자는 이 파일을 읽을 수만 있어야 합니다. 파일에는 암호 또는 토큰이 한 줄에 포함되어야 합니다.

다음 예를 사용하려면 *your_username*은 사용자 이름으로, *password.txt*는 암호 또는 토큰이 한 줄에 포함된 `.txt` 파일로, *image:tag*는 스캔할 이미지의 이름으로 바꿉니다.

```
INSPECTOR_SBOMGEN_USERNAME=your_username \
INSPECTOR_SBOMGEN_PASSWORD=`cat password.txt` \
./inspector-sbomgen container --image image:tag
```

Sbomgen의 예시 출력

다음은 Sbomgen을 사용하여 인벤토리에 추가된 컨테이너 이미지에 대한 SBOM의 예입니다.

컨테이너 이미지 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"
        }
      ]
    }
  ]
}
```

Amazon Inspector SBOM 생성기의 이전 버전

이 주제에서는 Amazon Inspector SBOM 생성기의 이전 버전에 대한 링크를 제공합니다. Sbomgen 설치에 대한 자세한 내용은 Sbomgen 설치를 참조하세요.

플랫폼	버전	SHA-256 체크섬
Linux AMD64 Linux ARM64	1.11.2 1.11.2	bef68671bc532e4fb529500b62d7af8360123cd967308d41ad0ce8f43f7762fb 4f11d7037efa443f442c4edf7ba28774c4fa706fb7622e4fba645bb3ad3958c9
Linux AMD64 Linux ARM64	1.11.1 1.11.1	809eb7cb80d24fbf6ffdd124438d53a907632c222e924913ebd61044ca949490 057f9e4c9970aeda4bda0685e7e02436fd5223fbe81cec65138551c63ed77ba0
Linux AMD64 Linux ARM64	1.11.0 1.11.0	5172a5556cf46f9fbc5cf1d35bd382919fb6b41aca1ec938db3a75530060b0cf c9e2da7b076dc89dc39a962a7dd9c7d1fd29230a4eec7eb95f951d6a179093d0
Linux AMD64	1.10.1	9e33622a7874adfe719ab7db75a1e44f4b5f

플랫폼	버전	SHA-256 체크섬	
Linux ARM64	1.10.1	ae3573374068b501c89f0accfcfe 78d5a7f800fc26ba86adab5b634431a91c007075e06d6ce46e50687d5156184e	
Linux AMD64 Linux ARM64	1.10.0 1.10.0	0b7a553d7d2d17c40a62f1a11013bc46fa2c3814f407c11130e15af3fe313769 5ce9e315a4f8f90ff5eed7ab058efc8dbff6593d66d3fc455f1c37e882ec6466	
Linux AMD64 Linux ARM64	1.9.1 1.9.1	d0ef4c14fec6c42e70ae55b3e44d17d027132947596e8ef861c0efc3c0e5a871 2d8145011c13f5611fc30f4510785d53e98b911717f6dbe69616af4d4b0df61f	

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.9.0 1.9.0	78b377b27 30eb15476 173e40885 454ae191e953663af3 e0928dddfb8608f465 5 985bdc06d25eccb87c 4a81995c8a2d3c78e1 c02beea309a620b2de 4954767591	
Linux AMD64 Linux ARM64	1.8.3 1.8.3	54eed5a772f68320f3 906bec5920e3a19da9 04abdace10f985b878 59015eef89 febd74a397fb0cdd33 56072503f08465ab87 2d1620d59 a2ab7d83bdb076c929 d	
Linux AMD64 Linux ARM64	1.8.2 1.8.2	2e4e3c754e23004634 9dd975feb48fa953ea 5a2de190cbbc17c1c8 5043936b5a 449a49e22 2a2bdffe0353435d7b 04b0556b35a391c7b9 714ce46d1a5382bc3e 2	

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.8.1 1.8.1	9ff7958e298d2b228b0c7617f0a9a873254587fc26aee9826c37273650b389e9 6737584fd2c7d24b56777d02846d1737f47d0121344baea217a3e5368fd98fcc	
Linux AMD64 Linux ARM64	1.8.0 1.8.0	ef32e7fb4ee0af1e47d6b528b47293fc7127c7a7539f7354e84452626a4c204d 0b82ddc691a517bb8fc6ccd67b80ca566b117a1bb410c05764c9b7e3ba76c510	
Linux AMD64 Linux ARM64	1.7.3 1.7.3	3fba95d44aaea55ad06d3c7635a671662c483474578376d3f11e84474f8de25f 1f4b52e3d80de87b92b563a78bac4a2d898e7af82db5b6791d899d516e97cfbb	

플랫폼	버전	SHA-256 체크섬
Linux AMD64 Linux ARM64	1.7.2 1.7.2	c44ba9bf1cfeb3ea2d6d0b15d25816800a5045a438474f2f77c390bac41ae4cb d37c5b1605bf82260da0b0f36311c83b1646a4327c3fd8169ba4b3a978470c9c
Linux AMD64 Linux ARM64	1.7.1 1.7.1	b0beb602a6ae439d4e307bd99682bc8a419fd7d5e78a278bfc718eb18e00b05e 95ff2d9df2fcd1982dd705df1e763f57a0b499b6fe06801e9a80869e2e464831
Linux AMD64 Linux ARM64	1.7.0 1.7.0	a6316c2ecd5fde7091d1099335f45f0e2400b3977c92ee4d72bd1eb359320e61 9751ba5e5c6c6c0aef7d29b1c4adbd4088da3a07bb77eaa7de3f04aa33ad8562

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.6.3 1.6.3	b6a309e87 9aaa78d7d 8e224eb5214df5fd41 5244d370885e6c8876 db5a4181d2 59ed0b7eb 7d1eadadb691f058d3 2634a03a856ba03ac2 ddb8cd3599ceb55cb9 a	
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	

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.6.0 1.6.0	ffe671c2c1d1c2142a4af056d1c179eaffbc3925f5afaa6f3d655bd495ce5e1c a733c0b00c7225369c68ad47c57846b4546e2c9f47580ab98394baefc765c134	
Linux AMD64 Linux ARM64	1.5.5 1.5.5	ebcfbe565631de5bc61b1d55d70a2d15b965f628678a2b60cffd01cd0c3443f1 a8e018ceee3a76dd4271f966015c216438b11ee807fcd970753e786baa335b56	
Linux AMD64 Linux ARM64	1.5.4 1.5.4	aa8c1ffacc563b87975497f53eddec0b29397a898fac19f4902b8acb7eeb347b c6ba98d441aa88d3d3150449c098cd13ce3baeccee45ad4c9a1326f8bb8f87fc	

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.5.3 1.5.3	d493c23121101c9c3df888e717bf81d7f7b81809754f3492e1ae52f02b089b68 8dfa5c97b3bd45da487706e95d1894290f53b113247bbb89b9fac16dab8184b6	
Linux AMD64 Linux ARM64	1.5.2 1.5.2	ff6233d7da9f7e963589a0eb8f07bee2ca375360365cb6b6e354585cf1371910 fd31efb6031754b2bc8414d7fe9dd14a067767704145af0559b3500cc437c7ee	
Linux AMD64 Linux ARM64	1.5.1 1.5.1	391fcc52117fed79cae6e92a9e225732166a6df2582aa7f6b5230149761f6732 f9bc90d18724f93db0f5ca3b79136adb7b49fa33fa179a5e87b4d512f256b56b	

플랫폼	버전	SHA-256 체크섬
Linux AMD64 Linux ARM64	1.5.0 1.5.0	d7b6cb84053358e462d76488d019140ecd05ad405217a60a96b727fb062880fe 067dcf5c302160a5270f89aed3f941bb0571dcb8a59f75dddb1b7747c2a82ec7
Linux AMD64 Linux ARM64	1.4.0 1.4.0	c8ca73761afd742e1deb98b04eb5714c9c2a574b652a763b18e23560e66aea24 188d9757782278653e65605aaf186feda104345ba2f9de438873e568f1ff6204
Linux AMD64 Linux ARM64	1.3.2 1.3.2	57dd5d135600e84690706cfe95860e78149988d37cf81429ce97b9256d179fb4 91526ecdafc6cc3718fabe75b2693ace5effb9c0af3327b484b7f5a154929997

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.3.1 1.3.1	097ec83907c459a36de11c92d016fffd64f1c33fd4bcbf2af465e0979b0d9237 aa93a3d402abc4a986a9ad9d3de8fcca81ee25a55596ac6dc4502ed1d6819502	
Linux AMD64 Linux ARM64	1.3.0 1.3.0	21439f92c314daf136832ca6676a65d288768aa69fc6dcd2014a3038b2701eeb 4a41779b0c3b32242eedef288de6c1bf40fda0d4246b32fd0cd8d4e51e58f94b	
Linux AMD64 Linux ARM64	1.2.1 1.2.1	e022e95e59f1790949bca8dbbb6478a5d3fb677ccd45aa4ba30ebd91ae86ad65 824acc5bb5b0210954fe9ab089d9461453a4975d34292cc0c676837c3a7279b4	

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.2.0 1.2.0	9625b1a8ae1937ca2179c2535a0ffceca934138e0b66feac9ba3e34ffaa22ec5 7f387e560b41571fb52efd9e620bf2b9e3a067ca781e88aaa977b2b8acdebf35	
Linux AMD64 Linux ARM64	1.1.1 1.1.1	6809b7e46675c66e3af354c53433dc46c4d1ddaf258e05ba15e38e784ea0285e 6361e59fb2448c66c4698ea33979ecaaefc2af4420034aabbbe741242f60dbdd	
Linux AMD64 Linux ARM64	1.1.0 1.1.0	f84c8815413d451490b38509950235f88713c0c61c7259a4831934995664bd8f aaffefb5e44195dc55d5fd3289e511720f64c130644cbd58103cf7f36e96f058	

플랫폼	버전	SHA-256 체크섬	
Linux AMD64 Linux ARM64	1.0.0 1.0.0	cc126e24962f1a6497cf17679b3e3b73be68963c47e3968a56e73caacf045b5c 5d5bf97a4acfeaaa73ad6c918738188e0c822e475ef37a334e49d77ba907b08a	

Amazon Inspector SBOM 생성기 포괄적인 운영 체제 컬렉션

Amazon Inspector SBOM 생성기는 다양한 운영 체제를 스캔하여 시스템 구성 요소에 대한 강력하고 상세한 분석을 보장합니다. SBOM을 생성하면 운영 체제의 구성을 이해하는 데 도움이 되므로 시스템 관리형 패키지의 취약성을 식별할 수 있습니다. 이 주제에서는 Amazon Inspector SBOM 생성기가 지원하는 다양한 운영 체제 패키지 컬렉션의 주요 특성에 대해 설명합니다. Amazon Inspector가 지원되는 인스턴스에 대한 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.

지원되는 운영 체제 아티팩트

Amazon Inspector SBOM 생성기는 다음 운영 체제 아티팩트를 지원합니다.

플랫폼	바이너리	소스	Stream
Alma Linux	해당 사항 없음	예	예
Alpine Linux	예	예	해당 사항 없음
Amazon Linux	해당 사항 없음	예	해당 사항 없음
CentOS	해당 사항 없음	예	해당 사항 없음
Chainguard	예	예	해당 사항 없음
Debian	예	예	해당 사항 없음

플랫폼	바이너리	소스	Stream
Distroless	예	예	해당 사항 없음
Fedora	해당 사항 없음	예	해당 사항 없음
MinimOS	예	예	해당 사항 없음
OpenSUSE	해당 사항 없음	예	해당 사항 없음
Oracle Linux	해당 사항 없음	예	해당 사항 없음
Photon OS	해당 사항 없음	예	해당 사항 없음
RHEL	해당 사항 없음	예	예
Rocky Linux	해당 사항 없음	예	예
SLES	해당 사항 없음	예	해당 사항 없음
Ubuntu	예	예	해당 사항 없음
Windows	해당 사항 없음	해당 사항 없음	해당 사항 없음

APK 기반 OS 패키지 컬렉션

이 섹션에는 APK 기반 OS 패키지 컬렉션에 지원되는 플랫폼과 주요 특성이 포함되어 있습니다. 자세한 내용은 Alpine Linux 웹 사이트의 Alpine Package Keeper를 참조하세요.

지원하는 플랫폼

지원되는 플랫폼은 다음과 같습니다.

- Alpine Linux

> **Note**
>
> APK 기반 시스템의 경우 Amazon Inspector SBOM 생성기는 `/lib/apk/db/` 파일에서 패키지 메타데이터를 수집합니다.

주요 기능

- 패키지 이름 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 버전 컬렉션 - 설치된 각 패키지의 버전을 추출합니다.
- 소스 패키지 식별 - 설치된 각 패키지의 소스 패키지를 식별합니다.

예제

다음 코드 조각은 APK 데이터베이스 파일의 예제입니다.

```
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 기반 OS 패키지 컬렉션

이 섹션에는 DPKG 기반 OS 패키지 컬렉션에 지원되는 플랫폼과 주요 특성이 포함되어 있습니다. 자세한 내용은 Debian 웹 사이트의 Debian 패키지를 참조하세요.

지원하는 플랫폼

다음 플랫폼을 지원합니다.

- Debian
- Ubuntu

> **Note**
>
> DPKG 기반 시스템의 경우 Amazon Inspector SBOM 생성기는 `/var/lib/dpkg/status` 파일에서 패키지 메타데이터를 수집합니다.

주요 기능

다음은 DPKG 기반 OS 패키지의 주요 특성입니다.

- 패키지 이름 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 버전 컬렉션 - 설치된 각 패키지의 버전을 추출합니다.
- 소스 패키지 식별 - 설치된 각 패키지의 소스 패키지를 식별합니다.

예제

다음 코드 조각은 `/var/lib/dpkg/` 파일의 예제입니다.

```
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 기반 OS 패키지 컬렉션

이 섹션에는 RPM 기반 OS 패키지 컬렉션에 지원되는 플랫폼과 주요 특성이 포함되어 있습니다. 자세한 내용은 RPM 웹 사이트에서 RPM Package Manager를 참조하세요.

지원하는 플랫폼

다음 플랫폼을 지원합니다.

- Alma Linux
- Amazon Linux
- CentOS
- Fedora
- OpenSUSE
- Oracle Linux
- PhotonOS
- RedHat Enterprise Linux
- Rocky Linux
- SUSE Linux Enterprise Server

> **Note**
>
> RPM 기반 시스템의 경우 Amazon Inspector SBOM 생성기는 `/var/lib/rpm` 파일에서 패키지 메타데이터를 수집합니다.

주요 기능

다음은 RPM 기반 OS 패키지 컬렉션의 주요 특성입니다.

- 패키지 이름 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 버전 컬렉션 - 설치된 각 패키지의 버전을 추출합니다.
- 소스 패키지 식별 - 설치된 각 패키지의 소스 패키지를 식별합니다.
- 스트림 지원 - 설치된 각 패키지의 스트림 메타데이터를 추출합니다.

예제

다음은 RPM 데이터베이스 파일 코드 조각의 예제입니다.

```
/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
```

Windows OS 버전 컬렉션

Linux 기반 운영 체제와 달리 Windows는 운영 체제 자체에 패키지 관리 시스템을 사용하지 않습니다. Amazon Inspector SBOM 생성기는 Windows OS 버전 정보만 수집합니다. Windows 애플리케이션 스캔의 경우 windows-apps 스캐너를 대신 사용합니다. windows-apps 스캐너는 Windows 시스템에 설치된 애플리케이션에 대한 정보를 수집합니다. 자세한 내용은 단원을 참조하십시오Microsoft applications 에코시스템 컬렉션.

주요 기능

- OS 버전 컬렉션 - Windows 레지스트리에서 Windows OS 버전을 추출합니다. 추출된 OS 버전은 Windows OS의 취약성 감지에 사용됩니다.

레지스트리 키 및 값

다음 Windows 레지스트리 키 및 값은 OS 이름 및 버전 정보를 수집하는 데 사용됩니다.

- 레지스트리 키

  ```
  HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
  ```

- 레지스트리 값
  - ProductName - OS 이름 및 에디션(예: "Windows Server 2025 Datacenter")
  - CurrentMajorVersionNumber - OS의 메이저 버전
  - CurrentMinorVersionNumber - OS의 마이너 버전
  - CurrentBuild - OS의 빌드 수

- UBR - OS의 개정 번호

Chainguard 이미지 패키지 컬렉션

이 섹션에는 Chainguard 이미지 패키지 컬렉션에 지원되는 플랫폼과 주요 특성이 포함되어 있습니다. 자세한 내용은 Chainguard 웹사이트의 이미지를 참조하세요.

지원하는 플랫폼

다음 플랫폼을 지원합니다.

- Wolfi Linux

> **Note**
>
> Chainguard 이미지의 경우 Amazon Inspector SBOM 생성기는 `/lib/apk/db/installed` 파일에서 패키지 메타데이터를 수집합니다.

주요 기능

주요 특성은 다음과 같습니다.

- 패키지 이름 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 버전 컬렉션 - 설치된 각 패키지의 버전을 추출합니다.
- 소스 패키지 식별 - 설치된 각 패키지의 소스 패키지를 식별합니다.

예제

다음 코드 조각은 Chainguard 이미지 파일의 예제입니다.

```
P:wolfi-keys
V:1-r8
A:x86_64
L:MIT
T:Wolfi signing keyring
```

```
o:wolfi-keys
```

Distroless 이미지 패키지 컬렉션

Distroless 컨테이너는 Linux 배포에서 패키지 관리자, 쉘 및 기타 유틸리티를 제외하는 컨테이너 이미지입니다. Distroless 컨테이너에는 애플리케이션을 실행하고 성능과 보안을 개선하는 데 필요한 필수 종속성만 포함됩니다.

> **Note**
>
> Distroless 이미지의 경우 Amazon Inspector SBOM 생성기는 `/var/lib/dpkg/status.d` 파일에서 패키지 메타데이터를 수집합니다. Debian 및 Ubuntu 기반 배포만 지원됩니다. "Debian" 또는 "Ubuntu"를 표시하는 `/etc/os-release` 파일 시스템의 `NAME` 필드로 식별할 수 있습니다.

주요 기능

- 패키지 이름 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 버전 컬렉션 - 설치된 각 패키지의 버전을 추출합니다.

예제

다음은 Distroless 이미지 파일의 예제입니다.

```
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 패키지 컬렉션

이 섹션에는 Minimus 이미지 패키지 컬렉션에 지원되는 플랫폼과 주요 특성이 포함되어 있습니다. 자세한 내용은 Minimus 웹사이트를 참조하세요.

지원하는 플랫폼

다음 플랫폼을 지원합니다.

- MinimOS

> **Note**
>
> Minimus 이미지의 경우 Amazon Inspector SBOM 생성기는 `/lib/apk/db/installed` 파일에서 패키지 메타데이터를 수집합니다.

주요 기능

주요 특성은 다음과 같습니다.

- 패키지 이름 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 버전 컬렉션 - 설치된 각 패키지의 이름을 추출합니다.
- 소스 패키지 식별 - 설치된 각 패키지의 소스 패키지를 식별합니다.

다음은 Minimus 이미지 파일의 코드 조각입니다.

```
P:ca-certificates-bundle
V:20241121-r1
A:aarch64
L:MPL-2.0 AND MIT
T:
```

```
o:ca-certificates
```

프로그래밍 언어 종속성 컬렉션

Amazon Inspector SBOM 생성기는 강력하고 상세한 종속성 모음을 구성하는 다양한 프로그래밍 언어 및 프레임워크를 지원합니다. SBOM을 생성하면 소프트웨어의 구성을 이해하는 데 도움이 되므로 취약성을 식별하고 보안 표준을 준수할 수 있습니다. Amazon Inspector SBOM 생성기는 다음과 같은 프로그래밍 언어 및 파일 형식을 지원합니다.

Go 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
Go	Go	go.mod	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
		go.sum	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
		Go Binaries	예	해당 사항 없음	해당 사항 없음	해당 사항 없음	아니요
		GOMODCACHE	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	

go.mod/go.sum

go.mod 및 go.sum 파일을 사용하여 Go 프로젝트에서 종속성을 정의하고 잠급니다. Amazon Inspector SBOM 생성기는 Go 도구 체인 버전에 따라 이러한 파일을 다르게 관리합니다.

주요 기능

- go.mod에서 종속성을 수집합니다(Go 도구 체인 버전이 1.17 이상인 경우).
- go.sum에서 종속성을 수집합니다(Go 도구 체인 버전이 1.17 이하인 경우).
- 선언된 모든 종속성 및 종속성 버전을 식별하기 위한 go.mod 구문 분석

예시 **go.mod** 파일

다음은 go.mod 파일의 예제입니다.

```
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
)
```

예시 **go.sum** 파일

다음은 go.sum 파일의 예제입니다.

```
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**
>
> 이러한 각 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Go 바이너리

Amazon Inspector SBOM 생성기는 컴파일된 Go 바이너리에서 종속성을 추출하여 사용 중인 코드를 보장합니다.

> **Note**
>
> Amazon Inspector SBOM 생성기는 공식 Go 컴파일러를 사용하여 구축된 Go 바이너리에서 도구 체인 버전을 캡처하고 평가할 수 있도록 지원합니다. 자세한 내용은 Go 웹 사이트에서 다운로드 및 설치를 참조하세요. Red Hat과 같은 다른 공급업체의 Go 도구 체인을 사용하는 경우 배포 및 메타데이터 가용성의 잠재적 차이로 인해 평가가 정확하지 않을 수 있습니다.

주요 기능

- Go 바이너리에서 직접 종속성 정보 추출
- 바이너리 내에 포함된 종속성을 수집합니다.
- 바이너리 컴파일에 사용되는 Go 도구 체인 버전을 감지하고 추출합니다.

GOMODCACHE

Amazon Inspector SBOM 생성기는 Go 모듈 캐시를 스캔하여 설치된 종속성에 대한 정보를 수집합니다. 이 캐시는 다운로드한 모듈을 저장하여 여러 빌드에서 동일한 버전이 사용되는지 확인합니다.

주요 기능

- `GOMODCACHE` 디렉터리를 스캔하여 캐시된 모듈 식별
- 모듈 이름, 버전 및 소스 URL을 포함한 자세한 메타데이터를 추출합니다.

구조 예제

다음은 `GOMODCACHE` 구조의 예입니다.

```
~/go/pkg/mod/
### github.com/gin-gonic/gin@v1.7.2
### golang.org/x/crypto@v0.0.0-20210616213533-5cf6c0f8e123
```

> **Note**
>
> 이 구조는 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Java 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
Java	Maven	컴파일된 Java 애플리케이션(.jar/.war/.ear) pom.xml	해당 사항 없음 해당 사항 없음	해당 사항 없음 해당 사항 없음	예 예	해당 사항 없음 해당 사항 없음	예 예

> **Note**
>
> 취약성 평가 특성은 Maven Central 리포지토리만 지원합니다. JBoss Enterprise Maven Repository와 같은 타사 리포지토리는 현재 지원되지 않습니다.

Amazon Inspector SBOM 생성기는 컴파일된 Java 애플리케이션 및 pom.xml 파일을 분석하여 Java 종속성 스캔을 수행합니다. 컴파일된 애플리케이션을 스캔할 때 스캐너는 무결성 확인을 위해 SHA–1 해시를 생성하고, 임베디드 pom.properties 파일을 추출하고, 중첩된 pom.xml 파일을 구문 분석합니다.

SHA–1 해시 컬렉션(컴파일된 .jar, .war, .ear 파일의 경우)

Amazon Inspector SBOM 생성기는 컴파일된 Java 아티팩트의 무결성과 추적성을 보장하기 위해 프로젝트의 모든 .ear, .jar 및 .war 파일에 대해 SHA–1 해시를 수집하려고 합니다.

주요 기능

- 모든 컴파일된 Java 아티팩트에 대해 SHA–1 해시를 생성합니다.

아티팩트 예

다음은 SHA–1 아티팩트의 예입니다.

```
{
  "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**
>
> 이 아티팩트는 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

pom.properties

pom.properties 파일은 Maven 프로젝트에서 패키지 이름 및 패키지 버전을 포함한 프로젝트 메타데이터를 저장하는 데 사용됩니다. Amazon Inspector SBOM 생성기는 이 파일을 구문 분석하여 프로젝트 정보를 수집합니다.

주요 기능

- 패키지 아티팩트, 패키지 그룹 및 패키지 버전을 구문 분석하고 추출합니다.

예시 **pom.properties** 파일

다음은 pom.properties파일의 예제입니다.

```
#Generated by Maven
#Tue Mar 16 15:44:02 UTC 2021

version=1.6.0
groupId=net.datafaker
artifactId=datafaker
```

> **Note**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

중첩 **pom.xml** 구문 분석 제외

컴파일된 Java 애플리케이션을 스캔할 때 pom.xml 구문 분석을 제외하려면 --skip-nested-pomxml 인수를 사용합니다.

pom.xml

pom.xml 파일은 Maven 프로젝트의 핵심 구성 파일입니다. 여기에는 프로젝트 및 프로젝트 종속성에 대한 정보가 포함되어 있습니다. Amazon Inspector SBOM 생성기는 pom.xml 파일을 구문 분석하여 종속성을 수집하여 리포지토리의 독립 실행형 파일과 컴파일된 .jar 파일 내의 파일을 스캔합니다.

주요 기능

- pom.xml 파일에서 패키지 아티팩트, 패키지 그룹 및 패키지 버전을 구문 분석하고 추출합니다.

지원되는 Maven 범위 및 태그

종속성은 다음 Maven 범위로 수집됩니다.

- 컴파일
- provided
- 런타임
- 테스트
- 시스템
- 가져오기

종속성은 Maven 태그 <optional>true</optional>을 사용하여 수집됩니다.

범위가 있는 **pom.xml** 파일 예제

다음은 범위가 있는 pom.xml 파일의 예제입니다.

```
<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>
```

범위가 없는 **pom.xml** 파일 예제

다음은 범위가 없는 pom.xml 파일의 예제입니다.

```
<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**
>
> 이러한 각 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

JavaScript 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
Javascript	Node Modules	node_modules/*/package.json	해당 사항 없음	해당 사항 없음	예	예	예
	NPM		해당 사항 없음	예	해당 사항 없음	해당 사항 없음	아니요
	PNPM			예	해당 사항 없음	해당 사항 없음	

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
	YARN	package-lock.json (v1, v2, and v3) / npm-shrinkwrap.json pnpm-lock.yaml yarn.lock	해당 사항 없음 해당 사항 없음	예	해당 사항 없음	해당 사항 없음	아니요 아니요

package.json

`package.json` 파일은 Node.js 프로젝트의 핵심 구성 요소입니다. 설치된 패키지에 대한 메타데이터가 포함되어 있습니다. Amazon Inspector SBOM 생성기는 이 파일을 스캔하여 패키지 이름과 패키지 버전을 식별합니다.

주요 기능

- JSON 파일 구조를 구문 분석하여 패키지 이름 및 버전 추출
- 프라이빗 값이 있는 프라이빗 패키지를 식별합니다.

예시 **`package.json`** 파일

다음은 `package.json`파일의 예제입니다.

```
{
"name": "arrify",
"private": true,
"version": "2.0.1",
"description": "Convert a value to an array",
"license": "MIT",
"repository": "sindresorhus/arrify"
}
```

> **Note**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

package-lock.json

`package-lock.json` 파일은 프로젝트에 설치된 종속성의 정확한 버전을 잠그기 위해 npm에 의해 자동으로 생성됩니다. 모든 종속성과 하위 종속성의 정확한 버전을 저장하여 환경에서 일관성을 보장합니다. 이 파일은 정규 종속성과 개발 종속성을 구별할 수 있습니다.

주요 기능

- JSON 파일 구조를 구문 분석하여 패키지 이름 및 패키지 버전을 추출합니다.
- 개발 종속성 감지 지원

예시 **`package-lock.json`** 파일

다음은 `package-lock.json`파일의 예제입니다.

```
"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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

npm-shrinkwrap.json

npm은 프로젝트에 설치된 종속성의 정확한 버전을 잠그기 위해 `package-lock.json` 및 `npm-shrinkwrap.json` 파일을 자동으로 생성합니다. 이렇게 하면 모든 종속성 및 하위 종속성의 정확한 버전을 저장하여 환경의 일관성을 보장할 수 있습니다. 파일은 정규 종속성과 개발 종속성을 구분합니다.

주요 기능

- JSON 파일 구조의 `package-lock` 버전 1, 2 및 3을 구문 분석하여 패키지 이름 및 버전 추출
- 개발자 종속성 감지가 지원됨(`package-lock.json`은 프로덕션 및 개발 종속성을 캡처하므로 도구가 개발 환경에서 사용되는 패키지를 식별할 수 있음)
- `npm-shrinkwrap.json` 파일이 `package-lock.json` 파일보다 우선함

예제

다음은 `package-lock.json`파일의 예제입니다.

```
"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

pnpm-lock.yaml 파일은 설치된 종속성 버전의 레코드를 유지하기 위해 pnpm에 의해 생성됩니다. 또한 개발 종속성을 별도로 추적합니다.

주요 기능

- YAML 파일 구조를 구문 분석하여 패키지 이름 및 버전 추출
- 개발 종속성 감지 지원

예제

다음은 pnpm-lock.yaml파일의 예제입니다.

```
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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

yarn.lock

Amazon Inspector SBOM 생성기는 컴파일된 Java 아티팩트의 무결성과 추적성을 보장하기 위해 프로젝트의 `.ear`, `.jar` 및 `.war` 파일에 대한 SHA–1 해시를 수집하려고 합니다.

주요 기능

- 모든 컴파일된 Java 아티팩트에 대해 SHA–1 해시를 생성합니다.

예제 SHA–1 아티팩트

다음은 SHA–1 아티팩트의 예입니다.

```
"@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**
>
> 이 아티팩트는 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

.NET 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
.NET	.NET Core	*.deps.json	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
	Nuget		해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
	Nuget .NET	Packages.config packages.lock.json .csproj	해당 사항 없음 해당 사항 없음	해당 사항 없음 해당 사항 없음	예 해당 사항 없음	해당 사항 없음 해당 사항 없음	예

Packages.config

Packages.config 파일은 이전 버전의 Nuget에서 프로젝트 종속성을 관리하는 데 사용하는 XML 파일입니다. 여기에는 특정 버전을 포함하여 프로젝트에서 참조하는 모든 패키지가 나열됩니다.

주요 기능

- XML 구조를 구문 분석하여 패키지 ID 및 버전 추출

예제

다음은 Packages.config파일의 예제입니다.

```
<?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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

*.deps.json

`*.deps.json` 파일은 .NET Core 프로젝트에서 생성되며 경로, 버전 및 런타임 종속성을 포함한 모든 종속성에 대한 자세한 정보를 포함합니다. 이 파일은 런타임에 올바른 버전의 종속성을 로드하는 데 필요한 정보가 있는지 확인합니다.

주요 기능

- 포괄적인 종속성 세부 정보를 위해 JSON 구조를 구문 분석합니다.
- `libraries` 목록에서 패키지 이름과 버전을 추출합니다.

예시 **`.deps.json`** 파일

다음은 `.deps.json`파일의 예제입니다.

```
{
"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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

packages.lock.json

`packages.lock.json` 파일은 최신 버전의 Nuget에서 .NET 프로젝트에 대한 종속성의 정확한 버전을 잠그는 데 사용되어 여러 환경에서 동일한 버전이 일관되게 사용되도록 합니다.

주요 기능

- JSON 구조를 구문 분석하여 잠긴 종속성을 나열
- 직접 종속성과 전이적 종속성 모두 지원
- 패키지 이름 및 확인된 버전을 추출

예시 **`packages.lock.json`** 파일

다음은 `packages.lock.json`파일의 예제입니다.

```
{
"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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

.csproj

`.csproj` 파일은 XML 및 .NET 프로젝트의 프로젝트 파일로 작성됩니다. 여기에는 Nuget 패키지, 프로젝트 속성 및 빌드 구성에 대한 참조가 포함됩니다.

주요 기능

- XML 구조를 구문 분석하여 패키지 참조를 추출합니다.

예시 **`.csproj`** 파일

다음은 .csproj파일의 예제입니다.

```
<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>
```

예시 **.csproj** 파일

다음은 .csproj파일의 예제입니다.

```
<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**
>
> 이러한 각 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

PHP 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
PHP	Composer	composer.lock	해당 사항 없음	해당 사항 없음	예	해당 사항 없음	예
		/ vendor/composer/installed.json	해당 사항 없음	해당 사항 없음	예	해당 사항 없음	예

composer.lock

composer 설치 또는 composer 업데이트 명령을 실행할 때 `composer.lock` 파일이 자동으로 생성됩니다. 이 파일은 모든 환경에 동일한 버전의 종속성이 설치되도록 보장합니다. 이는 일관되고 신뢰할 수 있는 빌드 프로세스를 제공합니다.

주요 기능

- 구조화된 데이터에 대한 JSON 형식 구문 분석
- 종속성 이름 및 버전 추출

예시 **composer.lock** 파일

다음은 `composer.lock`파일의 예제입니다.

```
{
"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**
>
> 그러면 패키지 URL이 포함된 출력이 생성됩니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

/vendor/composer/installed.json

`/vendor/composer/installed.json` 파일은 `vendor/composer` 디렉터리에 있으며 설치된 모든 패키지 및 패키지 버전의 포괄적인 목록을 제공합니다.

주요 기능

- 구조화된 데이터에 대한 JSON 형식 구문 분석
- 종속성 이름 및 버전 추출

예시 **`/vendor/composer/installed.json`** 파일

다음은 `/vendor/composer/installed.json`파일의 예제입니다.

```
{
```

```
"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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Python 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
Python	pip	requirements.txt	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
	Poetry	Poetry.lock	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
	Pipenv						예

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
	Egg/Wheel	Pipfile.lock	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
		.egg-info/PKG-INFO	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	
		.dist-info/METADATA	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	

requirements.txt

`requirements.txt` 파일은 Python 프로젝트에서 프로젝트 종속성을 지정하는 데 널리 사용되는 형식입니다. 이 파일의 각 줄에는 버전 제약 조건이 있는 패키지가 포함되어 있습니다. Amazon Inspector SBOM 생성기는 이 파일을 구문 분석하여 종속성을 정확하게 식별하고 카탈로그화합니다.

주요 기능

- 버전 지정자(== 및 "=) 지원
- 설명 및 복잡한 종속성 라인 지원

> **Note**
>
> 버전 지정자 <= 및 =>는 지원되지 않습니다.

예시 **`requirements.txt`** 파일

다음은 `requirements.txt`파일의 예제입니다.

```
flask==1.1.2
requests==2.24.0
numpy==1.18.5
foo~=1.2.0
# Comment about a dependency
scipy. # invalid
```

> **Note**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Pipfile.lock

Pipenv는 모든 패키징 환경(번들, 고정, 고정 해제)의 장점을 모두 제공하는 도구입니다. `Pipfile.lock`은 정확한 버전의 종속성을 잠가 결정적 빌드를 용이하게 합니다. Amazon Inspector SBOM 생성기는 이 파일을 읽어 종속성과 확인된 버전을 나열합니다.

주요 기능

- 종속성 확인을 위해 JSON 형식 구문 분석
- 기본 및 개발 종속성 지원

예시 **`Pipfile.lock`** 파일

다음은 `Pipfile.lock`파일의 예제입니다.

```
{
"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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Poetry.lock

Poetry는 Python의 종속성 관리 및 패키징 도구입니다. `Poetry.lock` 파일은 정확한 버전의 종속성을 잠가 일관된 환경을 용이하게 합니다. Amazon Inspector SBOM 생성기는 이 파일에서 자세한 종속성 정보를 추출합니다.

주요 기능

- 구조화된 데이터에 대한 TOML 형식 구문 분석
- 종속성 이름 및 버전 추출

예시 **`Poetry.lock`** 파일

다음은 `Poetry.lock`파일의 예제입니다.

```
[[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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

에그/휠

전 세계에 설치된 Python 패키지의 경우 Amazon Inspector SBOM 생성기는 `.egg-info/PKG-INFO` 및 `.dist-info/METADATA` 디렉터리에 있는 메타데이터 파일 구문 분석을 지원합니다. 이러한 파일은 설치된 패키지에 대한 자세한 메타데이터를 제공합니다.

주요 기능

- 패키지 이름 및 버전 추출
- 에그 및 휠 형식 모두 지원

예시 **PKG-INFO/METADATA** 파일

다음은 `PKG-INFO/METADATA`파일의 예제입니다.

```
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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Ruby 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
Ruby	Bundler	Gemfile.lock	해당 사항 없음	해당 사항 없음	예	해당 사항 없음	예
		.gemspec	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
		globall installed Gems	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	

Gemfile.lock

`Gemfile.lock` 파일은 모든 종속성의 정확한 버전을 잠가 모든 환경에서 동일한 버전이 사용되는지 확인합니다.

주요 기능

- ID 종속성 및 종속성 버전에 `Gemfile.lock` 파일을 구문 분석
- 세부 패키지 이름 및 패키지 버전 추출

예시 **`Gemfile.lock`** 파일

다음은 `Gemfile.lock`파일의 예제입니다.

```
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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

.gemspec

`.gemspec` 파일은 gem에 대한 메타데이터가 포함된 RubyGem 파일입니다. Amazon Inspector SBOM 생성기는 이 파일을 구문 분석하여 gem에 대한 자세한 정보를 수집합니다.

주요 기능

- gem 이름 및 gem 버전을 구문 분석하고 추출

> **Note**
>
> 참조 사양은 지원되지 않습니다.

예시 **`.gemspec`** 파일

다음은 `.gemspec`파일의 예제입니다.

```
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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

전역적으로 설치된 gem

Amazon Inspector SBOM 생성기는 Amazon EC2/Amazon ECR의 `/usr/local/lib/ruby/gems/<ruby_version>/gems/` 및 Lambda의 `ruby/gems/<ruby_version>/gems/`와 같은 표준 디렉터리에 있는 전역적으로 설치된 gem 스캔을 지원합니다. 이렇게 하면 전역적으로 설치된 모든 종속성을 식별하고 카탈로그화할 수 있습니다.

주요 기능

- 표준 디렉터리에서 전역적으로 설치된 모든 gem을 식별하고 스캔합니다.
- 전역적으로 설치된 각 gem에 대한 메타데이터 및 버전 정보를 추출

디렉터리 구조 예제

다음은 디렉터리 구조의 예제입니다.

```
.
### /usr/local/lib/ruby/3.5.0/gems/
### actrivesupport-6.1.4
### concurrent-ruby-1.1.9
### i18n-1.8.10
```

> **Note**
>
> 이 구조는 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Rust 종속성 스캔

프로그래밍 언어	패키지 관리자	지원되는 아티팩트	도구 체인 지원	개발 종속성	전이적 종속성	프라이빗 플래그	재귀적
Rust	Cargo.toml	Cargo.toml	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	예
		Cargo.lock	해당 사항 없음	해당 사항 없음	예	해당 사항 없음	예
		Rust binary (built with cargo-auditable)	예	해당 사항 없음	해당 사항 없음	해당 사항 없음	예

Cargo.toml

`Cargo.toml` 파일은 Rust 프로젝트의 매니페스트 파일입니다.

주요 기능

- `Cargo.toml` 파일을 구문 분석하고 추출하여 프로젝트 패키지 이름과 버전을 식별합니다.

예시 **`Cargo.toml`** 파일

다음은 `Cargo.toml`파일의 예제입니다.

```
[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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

Cargo.lock

`Cargo.lock` 파일은 종속성 버전을 잠가 프로젝트가 빌드될 때마다 동일한 버전이 사용되도록 합니다.

주요 기능

- `Cargo.lock` 파일을 구문 분석하여 모든 종속성 및 종속성 버전을 식별합니다.

예시 **`Cargo.lock`** 파일

다음은 `Cargo.lock`파일의 예제입니다.

```
# 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**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

cargo 감사 가능한 Rust 바이너리

Amazon Inspector SBOM 생성기는 `cargo-auditable` 라이브러리로 빌드된 Rust 바이너리에서 종속성을 수집합니다. 이렇게 하면 컴파일된 바이너리에서 종속성 추출을 활성화하여 추가 종속성 정보를 제공합니다.

주요 기능

- `cargo-auditable` 라이브러리로 빌드된 Rust 바이너리에서 직접 종속성 정보를 추출합니다.
- 바이너리에 포함된 종속성에 대한 메타데이터 및 버전 정보를 검색합니다.

> **Note**
>
> 이 파일은 패키지 URL이 포함된 출력을 생성합니다. 이 URL은 소프트웨어 재료표를 생성할 때 소프트웨어 패키지에 대한 정보를 지정하는 데 사용할 수 있으며 ScanSbom API에 포함될 수 있습니다. 자세한 내용은 GitHub 웹 사이트의 package-url 섹션을 참조하세요.

지원되지 않는 아티팩트

이 섹션에서는 지원되지 않는 아티팩트에 대해 설명합니다.

Java

Amazon Inspector SBOM 생성기는 메인스트림 Maven 리포지토리에서 가져온 종속성에 대한 취약성 감지만 지원합니다. Red Hat Maven 및 Jenkins와 같은 프라이빗 또는 사용자 지정 Maven 리포지토리는 지원되지 않습니다. 취약성을 정확하게 감지하려면 Java 종속성을 메인스트림 Maven 리포지토리에서 가져와야 합니다. 다른 리포지토리의 종속성은 취약성 스캔에서 다루지 않습니다.

JavaScript

esbuild 번들

esbuild 축소된 번들의 경우 Amazon Inspector SBOM 생성기는 esbuild를 사용하는 프로젝트에 대한 종속성 스캔을 지원하지 않습니다. esbuild에서 생성된 소스 맵에는 정확한 Sbomgen 생성에 필요한 충분한 메타데이터(종속성 이름 및 버전)가 포함되어 있지 않습니다. 안정적인 결과를 얻으려면 번들링 프로세스 전에 `node_modules/directory` 및 `package-lock.json`과 같은 원본 프로젝트 파일을 스캔합니다.

package.json

Amazon Inspector SBOM 생성기는 루트 수준 package.json 파일의 종속성 정보 스캔을 지원하지 않습니다. 이 파일은 패키지 이름 및 버전 범위만 지정하지만 완전히 확인된 패키지 버전은 포함하지 않습니다. 정확한 스캔 결과를 얻으려면 해결된 버전이 포함된 `package.json` 또는 `yarn.lock` 또는 `pnpm.lock` 같은 기타 잠금 파일을 사용합니다.

Dotnet

`PackageReference`에서 부동 버전 또는 버전 범위를 사용하는 경우 패키지 확인을 수행하지 않고 프로젝트에 사용되는 정확한 패키지 버전을 확인하는 것이 더 어려워집니다. 유동 버전 및 버전 범위를 통해 개발자는 고정 버전 대신 허용 가능한 패키지 버전 범위를 지정할 수 있습니다.

Go 바이너리

Amazon Inspector SBOM 생성기는 빌드 ID를 제외하도록 구성된 빌드 플래그로 빌드된 Go 바이너리를 스캔하지 않습니다. 이러한 빌드 플래그는 Amazon Inspector SBOM 생성기가 바이너리를 원래 소스에 정확하게 매핑하지 못하도록 합니다. 패키지 정보를 추출할 수 없기 때문에 불분명한 Go 바이너리가 지원되지 않습니다. 정확한 종속성 스캔을 위해 Go 바이너리가 빌드 ID를 포함한 기본 설정으로 빌드되었는지 확인합니다.

Rust 바이너리

Amazon Inspector SBOM 생성기는 Rust 바이너리가 cargo 감사 가능 라이브러리를 사용하여 빌드된 경우에만 바이너리를 스캔합니다. 이 라이브러리를 사용하지 않는 Rust 바이너리에는 정확한 종속성 추출에 필요한 메타데이터가 없습니다. Amazon Inspector SBOM 생성기는 Rust 1.7.3부터 시작하지만 Linux 환경의 바이너리에 대해서만 컴파일된 Rust 도구 체인 버전을 추출합니다. 포괄적인 스캔을 위해 cargo 감사 가능을 사용하여 Linux에 Rust 바이너리를 구축합니다.

> **Note**
>
> Rust 도구 체인 버전이 추출되더라도 도구 체인 자체에 대한 취약성 감지는 지원되지 않습니다.

Amazon Inspector SBOM 생성기 포괄적인 에코시스템 컬렉션

Amazon Inspector SBOM 생성기는 소프트웨어 재료표(SBOM)를 생성하고 운영 체제 및 프로그래밍 언어에서 지원되는 패키지에 대한 취약성 스캔을 수행하는 도구입니다. 핵심 운영 체제를 넘어 다양한 에코시스템의 스캔을 지원하여 인프라 구성 요소에 대한 강력하고 상세한 분석을 보장합니다. SBOM을 생성하면 최신 기술 스택의 구성을 이해하고, 에코시스템 구성 요소의 취약성을 식별하고, 타사 소프트웨어에 대한 가시성을 확보할 수 있습니다.

지원되는 에코시스템

에코시스템 컬렉션은 OS 패키지 관리자를 통해 설치된 패키지 이상으로 SBOM 생성을 확장합니다. 이는 수동 설치와 같은 대체 방법에 배포된 애플리케이션 모음을 통해 수행됩니다. Amazon Inspector SBOM 생성기는 다음 에코시스템에 대한 스캔을 지원합니다.

에코시스템	애플리케이션
7-Zip	7-Zip 아카이버(버전 21.07 이상)

에코시스템	애플리케이션
Apache	Apache httpd Apache tomcat
Atlassian	Jira Core Confluence Jira Software Jira Service Management
Curl	Curl Libcurl
Elasticsearch	Elasticsearch
Google	Chrome
Java	JDK JRE Amazon Corretto
Jenkins	Jenkins (버전 2.400.* 이상)
MariaDB 및 MySQL	MariaDB Server (10.6+, 11.x, 12.x) Oracle MySQL Server Server (8.0, 8.4, 9.4 이상)

에코시스템	애플리케이션
Microsoft applications	PowerShell NuGet CLI Visual Studio Code Microsoft Edge SharePoint Server Microsoft Defender Exchange Server Visual Studio .NET Runtime ASP.NET Core Runtime Microsoft Teams Outlook for Windows Microsoft Office Microsoft 365
Nginx	Nginx
Node	Node
Node.JS	node
OpenSSH	OpenSSH (버전 9 및 10)
OpenSSL	OpenSSL
Oracle	Oracle Database Server
PHP	PHP (버전 8.1 이상)

에코시스템	애플리케이션
WordPress	core plugin theme

7-Zip 에코시스템 컬렉션

지원되는 애플리케이션

- 7 Zip 아카이버(버전 21.07 이상)

주요 기능

- 7-Zip 바이너리를 검사하여 임베디드 버전 정보를 추출합니다.

> **Note**
> 특히 바이너리에서 제품 버전 값을 검색합니다.

지원되는 플랫폼 - 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`

예제 PURL

다음은 7-Zip의 패키지 URL 예제입니다.

```
pkg:generic/7zip/7zip@25.01
```

Apache 에코시스템 컬렉션

이 섹션에서는 Apache httpd 및 Apache tomcat 애플리케이션에 대한 세부 정보를 제공합니다.

Apache httpd

지원되는 애플리케이션

- Apache httpd

> **Note**
>
> 취약성 평가는 Apache httpd 버전 2.0 이상에만 적용됩니다.

주요 기능

- `/include/ap_release.h` 파일을 구문 분석하여 메이저 식별자 문자열, 마이너 식별자 문자열 및 패치 식별자 문자열이 포함된 설치 매크로를 추출합니다.

지원하는 플랫폼

Amazon Inspector SBOM 생성기는 플랫폼 전반의 공통 설치 경로에서 설치를 스캔합니다.

Unix

- `/usr/local/apache2/include/`

Windows

- `/Apache24/include/`
- `/Program Files/Apache24/include/`
- `/Program Files (x86)/Apache24/include/`

예시 **ap_release.h** 파일

다음은 ap_release.h 파일의 내용을 보여주는 예제입니다.

```
//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
```

예제 PURL

다음은 Apache httpd 애플리케이션의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/apache/httpd@2.4.1
```

Apache tomcat

지원되는 애플리케이션

- Apache tomcat

> **Note**
>
> 취약성 평가는 Apache tomcat 버전 9.0 이상에만 적용됩니다.

주요 기능

- `catalina.jar` 파일 압축을 풀어 `META-INF/MANIFEST.MF` 파일 내에서 버전 문자열이 포함된 설치 매크로를 추출합니다.

지원하는 플랫폼

Amazon Inspector SBOM 생성기는 플랫폼 전반의 공통 설치 경로에서 설치를 스캔합니다.

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/`

예시 **catalina.jar/META-INF/MANIFEST.MF** 파일

다음은 `catalina.jar/META-INF/MANIFEST.MF` 파일의 내용을 보여주는 예제입니다.

```
//truncated

Implementation-Title: Apache Tomcat
Implementation-Vendor: Apache Software Foundation
Implementation-Version: 10.1.31

//truncated
```

예제 PURL

다음은 Apache tomcat 애플리케이션의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/apache/tomcat@10.1.31
```

Atlassian 에코시스템 컬렉션

이 섹션에서는 Atlassian 서버 제품 및 애플리케이션에 대한 세부 정보를 제공합니다.

Atlassian Server Products

지원되는 애플리케이션

- Jira Core
- Confluence

주요 기능

- Jira Core -에서 Maven POM 속성을 구문 분석atlassian-jira-webapp하여 버전 정보를 추출합니다.
- Confluence -에서 Maven POM 속성을 구문 분석confluence-webapp하여 버전 정보를 추출합니다.

지원하는 플랫폼

Amazon Inspector SBOM 생성기는 공통 설치 경로에서 설치를 스캔합니다.

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

예제 PURL

다음은 Atlassian 서버 제품의 패키지 URLs.

```
// Jira Core
pkg:generic/atlassian/jira-core@10.0.1?distro=linux

// Confluence
pkg:generic/atlassian/confluence@9.2.7?distro=linux
```

Atlassian Applications

지원되는 애플리케이션

- Jira Software
- Jira Service Management

주요 기능

- Jira Software - `jira-software-application` JAR을 통해 감지하고 Maven POM 속성에서 버전을 추출합니다.
- Jira Service Management - `jira-servicedesk-application` JAR을 통해 감지하고 Maven POM 속성에서 버전을 추출합니다.

지원하는 플랫폼

Amazon Inspector SBOM 생성기는 공통 설치 경로에서 설치를 스캔합니다.

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`

예제 PURL

다음은 Atlassian 애플리케이션의 패키지 URLs.

```
// 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
```

Curl 에코시스템 컬렉션

이 섹션에서는 Curl 및 Libcurl 애플리케이션에 대한 세부 정보를 제공합니다.

Curl

지원되는 애플리케이션

- Curl

지원하는 플랫폼

- Unix – Linux 및 macOS
  - /usr/local/bin/curl

주요 기능 - Curl

- curl 바이너리를 검사하여 임베디드 버전 정보를 추출합니다.

> **Note**
>
> 특히 이진 실행 파일 `.rodata` 섹션(Linux의 ELF 바이너리용), `.rdata` 섹션(Windows의 PE 바이너리용) 또는 __cstring 섹션(macOS의 MachO 바이너리용)에서 버전 문자열을 검색합니다.

Curl version string

다음은 Curl 바이너리에 포함된 버전 문자열의 예입니다.

```
curl/8.14.1
```

버전을 식별하기 위해 문자열에서 `Curl` 버전이 `8.14.1` 추출됩니다.

예제 PURL(Curl)

다음은 `Curl` 버전 파일의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/curl/curl@8.14.1
```

Libcurl

지원되는 애플리케이션

- Libcurl

지원하는 플랫폼

- Unix – Linux 및 macOS
  - /usr/local/bin/curl/curlver.h

주요 기능 - Libcurl

- 를 검사curlver.h하여에 대한 임베디드 버전 정보를 추출합니다Libcurl.

> **Note**
>
> 특히 정의된 , `LIBCURL_VERSION_MAJOR` `LIBCURL_VERSION_MINOR`및 `LIBCURL_VERSION_PATCH` 변수에서 버전을 추출합니다.

Libcurl version string

다음은 `curlver.h` 파일의 버전 변수 예제입니다.

```
#define LIBCURL_VERSION_MAJOR 8
    #define LIBCURL_VERSION_MINOR 14
    #define LIBCURL_VERSION_PATCH 1
```

버전`8.14.1`은 이러한 줄에서 추출되어 `Libcurl` 버전을 식별합니다.

예제 PURL(Libcurl)

다음은 `Libcurl` 버전 파일의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/curl/libcurl@8.14.1
```

Elasticsearch 에코시스템 컬렉션

지원되는 애플리케이션

- Elasticsearch

> **Note**
>
> 취약성 평가는 Elasticsearch 버전 7.17.0에만 적용됩니다.

주요 기능

- Version - `elasticsearch-<specific.version>.jar` 파일의 압축을 풀어 Elasticsearch 버전 문자열이 포함된 `META-INF/MANIFEST.MF` 파일 내부의 설치 매크로를 추출합니다.

지원하는 플랫폼

- Linux – `/etc/elasticsearch/lib`, `/opt/elasticsearch/lib/`및 `/usr/share/elasticsearch/lib/`
- macOS – `/usr/local/var/lib/elasticsearch/lib/`
- Windows – `/elasticsearch/`, `/Program Files (x86)/Elastic/elasticsearch/lib/`및 `/Program Files/Elastic/elasticsearch/lib/`

예시 **`elasticsearch-<specific.version>.jar/META-INF/MANIFEST.MF`** 파일

다음은 `elasticsearch-<specific.version>.jar/META-INF/MANIFEST.MF` 파일의 예입니다.

```
//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
```

예제 PURL

다음은 `elasticsearch-<specific.version>.jar/META-INF/MANIFEST.MF` 파일의 패키지 URL 예제입니다.

```
pkg:generic/elastic/elasticsearch@8.19.0-SNAPSHOT
```

Google 에코시스템 컬렉션

지원되는 애플리케이션

- Google Chrome
- Puppeteer (puppeteer 라이브러리 지원, puppeteer-core는 포함되지 않음)

> **Note**
>
> Puppeteer는 puppeteer 라이브러리를 지원합니다. Puppeteer 코어는 포함되지 않습니다.

지원되는 아티팩트

Amazon Inspector는 다음에서 Google Chrome 정보를 수집합니다.

- `chrome/VERSION` 파일(빌드 소스)
- `chrome.exe` 파일(Windows Chrome 설치)
- `puppeteer` 파일(설치)

지원되는 각 아티팩트에 대해 Sbomgen은 chrome 파일 또는 puppeteer 파일을 구문 분석하고 수집합니다. puppeteer 설치의 경우 해당 Chromium 버전은 puppeteer 버전에 따라 수집됩니다. 자세한 내용은 Puppeteer 웹 사이트에서 지원되는 브라우저를 참조하세요.

`PUPPETEER_SKIP_CHROMIUM_DOWNLOAD` 환경 변수가 `true`로 설정되면 평가를 건너뛰고 `skip_chromium_download=true` 한정자가 Puppeteer 패키지 URL에 추가됩니다.

`chrome/VERSION` 버전 파일 예제

다음은 chrome/VERSION 버전 파일의 예입니다.

```
MAJOR=130
MINOR=0
BUILD=6723
PATCH=58
```

예제 PURL

다음은 chrome/VERSION 버전 파일의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/google/chrome@131.0.6778.87
```

puppeteer 버전 파일 예제

다음은 puppeteer 버전 파일의 예입니다.

```
{
"name": "puppeteer",
"version": "23.9.0",
"description": "A high-level API to control headless Chrome over the DevTools
 Protocol",
"keywords": [
  "puppeteer",
  "chrome",
  "headless",
  "automation"
]
}
```

예제 PURL

다음은 puppeteer 버전 파일의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/google/puppeteer@23.9.0
```

예제 PURL

다음은 puppeteer 버전 파일에 대한 건너뛰기 한정자가 있는 패키지 URL의 예입니다.

```
pkg:generic/google/puppeteer@22.15.0?distro=linux&skip_chromium_download=true
```

Java 에코시스템 컬렉션

지원되는 애플리케이션

- Oracle JDK
- Oracle JRE
- Amazon Corretto

주요 기능

- Java 설치 문자열을 추출합니다.
- Java 런타임이 포함된 디렉터리 경로를 식별합니다.
- 공급업체를 Oracle JDK, Oracle JRE 및 Amazon Corretto로 식별합니다.

Amazon Inspector SBOM 생성기는 다음 설치 경로 및 플랫폼에서 Java 설치를 스캔합니다.

- macOS: `/Library/Java/JavaVirtualMachines`
- Linux 32-bit: `/usr/lib/jvm`
- Linux 64-bit: `/usr/lib64/jvm`
- Linux (generic): `/usr/java and /opt/java`

Java 버전 정보의 예

다음은 Oracle Java 릴리스의 예입니다.

```
// 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"
```

예제 PURL

다음은 Oracle Java 릴리스의 패키지 URL 예제입니다.

```
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 에코시스템 컬렉션

지원되는 애플리케이션

- Jenkins 코어

> **Note**
>
> 취약성 평가는 Jenkins 버전 2.400.* 이상에 적용됩니다.

주요 기능

- 버전 문자열이 포함된 `META-INF/MANIFEST.M` 파일을 읽어 `jenkins.war` 파일에서 Jenkins 버전 정보를 추출합니다.

Amazon Inspector SBOM 생성기는 플랫폼 전반의 공통 설치 경로에서 Jenkins 설치를 찾습니다.

Linux

- `/usr/share/jenkins/jenkins.war`
- /usr/share/java/jenkins.war

macOS

- `/opt/homebrew/opt/jenkins-lts/libexec/jenkins.war`

Windows

- `/Program Files/Jenkins/Jenkins.war`
- `/Program Files (x86)/Jenkins/Jenkins.war`

예제 파일

다음은 다양한 릴리스에 대한 `jenkins.war/META-INF/MANIFEST.MF` 파일의 예입니다.

```
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
```

샘플 PURLs

다음은 Jenkins LTS 릴리스 버전 2.516.2 및 Jenkins 자동화 서버 릴리스 버전 2.414의 패키지 URLs입니다.

```
LTS: pkg:generic/jenkins/jenkins-core-lts@2.516.2.1
Regular: pkg:generic/jenkins/jenkins-core@2.414
```

MariaDB 및 MySQL 에코시스템 컬렉션

MariaDB

지원되는 애플리케이션

- MariaDB Server (10.6+, 11.x, 12.x)

주요 기능

- 데이터베이스별 패턴을 사용하여 데이터베이스 서버 바이너리 및 헤더 파일에서 버전 정보를 추출합니다.

- 데이터베이스 서버 설치가 포함된 디렉터리 경로를 식별합니다.
- 데이터 기반 파일 유형 감지를 사용하여 MariaDB와 MySQL 설치를 자동으로 구분합니다.

SBOM 생성기는 플랫폼 전반의 공통 MariaDB 설치 경로에서 설치를 찾습니다.

Linux

- `/usr/bin/mariadbd`
- `/usr/sbin/mariadbd`
- `/usr/local/bin/mariadbd`

macOS

- `C:/Program Files (x86)/MariaDB/include/mysql/mariadb_version.h (MariaDB)`
- `C:/Program Files/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)`

예제 PURL

다음은 MariaDB 서버의 패키지 URL 예제입니다.

```
# MariaDB Server

pkg:generic/mysql/mariadb-server@10.11.8
```

MySQL 에코시스템 컬렉션

지원되는 애플리케이션

- Oracle MySQL Server Server (8.0, 8.4, 9.4 이상)

주요 기능

- 데이터베이스별 패턴을 사용하여 데이터베이스 서버 바이너리 및 헤더 파일에서 버전 정보를 추출합니다.
- 데이터베이스 서버 설치가 포함된 디렉터리 경로를 식별합니다.
- 데이터 기반 파일 유형 감지를 사용하여 MySQL와 MariaDB 설치를 자동으로 구분합니다.

SBOM 생성기는 플랫폼 전반의 공통 MySQL 설치 경로에서 설치를 찾습니다.

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)`

예제 PURL

다음은 MySQL 서버의 패키지 URL 예제입니다.

```
# Oracle MySQL Server

pkg:generic/mysql/mysql-server@8.0.43
```

Microsoft applications 에코시스템 컬렉션

지원되는 Microsoft 애플리케이션

- PowerShell

- NuGet CLI
- Visual Studio Code
- Microsoft Edge
- SharePoint Server
- Microsoft Defender
- Exchange Server
- Visual Studio
- .NET Runtime
- ASP.NET Core Runtime
- Microsoft Teams
- Outlook for Windows
- Microsoft Office
- Microsoft 365

주요 기능

- PowerShell - `pwsh.exe` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- NuGet CLI - `nuget.exe` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- Visual Studio Code - `Code.exe` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- Microsoft Edge - `msedge.exe` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- SharePoint Server - `Microsoft.SharePoint.dll` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- Microsoft Defender - `MsMpEng.exe` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- Exhange Server - `Exsetup.exe` 파일을 검사하여 임베디드 버전 정보를 추출합니다.
- Visual Studio - `state.json` 파일을 구문 분석하여 `catalogInfo.productDisplayVersion` 필드에서 버전 문자열을 검색합니다.
- .NET Runtime - 설치 경로에서 `Microsoft.NETCore.App.deps.json` 파일을 검색하고 다음 파일 경로 패턴에서 버전 문자열을 추출합니다.

```
Microsoft.NETCore.App/<VERSION>/Microsoft.NETCore.App.deps.json
```

- ASP.NET Runtime - 설치 경로에서 `Microsoft.AspNetCore.App.deps.json` 파일을 검색하고 다음 파일 경로 패턴에서 버전 문자열을 추출합니다.

```
Microsoft.AspNetCore.App/<VERSION>/Microsoft.AspNetCore.App.deps.json
```

- Outlook for Windows - Windows 레지스트리를 구문 분석하고 다음 레지스트리 키에서 버전을 추출합니다.

```
HKLM\SOFTWARE\Classes\Local Settings\Software\Microsoft
\Windows\CurrentVersion\AppModel\PackageRepository\Packages
\Microsoft.OutlookForWindows_<VERSION>_<ARCH>__8wekyb3d8bbwe
```

- Microsoft Teams - Windows 레지스트리를 구문 분석하고 다음 레지스트리 키에서 버전을 추출합니다.

```
HKLM\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion
\AppModel\PackageRepository\Packages\MSTeams_<VERSION>_<ARCH>__8wekyb3d8bbwee
```

- Microsoft Office 365 / Microsoft 365 - Windows 레지스트리를 구문 분석하고 다음 레지스트리 키 및 값에서 버전을 추출합니다.
  - 레지스트리 키

    ```
    KEY_LOCAL_MACHINES\SOFTWARE\Microsoft\Office\ClickToRun\Configuration
    ```

  - 레지스트리 값
    - VersionToReport - Microsoft Office 버전
    - ProductReleaseIds - 제품 IDs. 설치된 Office 제품을 식별하는 데 사용됩니다. 제품 IDsproduct IDs에 대한 자세한 내용은 Microsoft 웹 사이트의 섹션을 참조하세요.
- Microsoft Office Suite - 다음 실행 파일을 검사하여 설치된 각 Office 애플리케이션을 수집합니다.
  - `EXCEL.EXE` – Microsoft Excel
  - `WINWORD.EXE` – Microsoft Word
  - `POWERPNT.EXE` – Microsoft PowerPoint
  - `OUTLOOK.EXE` – Microsoft Outlook

  Windows 레지스트리의 버전 번호는 설치된 각 Office 애플리케이션의 신뢰할 수 있는 버전 번호로 사용됩니다.

예시 **`state.json`** 파일

다음은 설치된 Visual Studio 버전을 수집하는 데 사용할 `state.json` 파일의 예입니다.

```
{
    "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
```

예제 PURL

다음은 각에 대한 패키지 URL의 예입니다Microsoft Applications.

```
// 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 Runtime
Sample PURL: pkg:generic/microsoft/dotnet@8.0.18

// 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
```

Nginx 에코시스템 컬렉션

지원되는 애플리케이션

- Nginx

지원하는 플랫폼

지원되는 플랫폼은 다음과 같습니다.

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.z는 임의 버전)

macOS

- /usr/local/etc/nginx/nginx

주요 기능

이 컬렉션은 바이너리를 검사하여 임베디드 버전 정보를 추출합니다. 바이너리 실행 파일 `.rodata` 섹션(Linux의 ELF 바이너리), `.rdata` 섹션(Windows의 PE 바이너리) 또는 `__ctring` 섹션(MachO 바이너리)에서 버전 문자열을 검색합니다.

버전 문자열 예

다음은 Nginx 바이너리에 포함된 버전 문자열의 예입니다.

```
nginx version: nginx/1.27.5
```

Nginx 버전을 식별하기 위해 `1.27.5` 버전이 추출됩니다.

예제 PURL

다음은 Nginx의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/nginx/nginx@1.27.5
```

Node.JS 런타임 컬렉션

지원되는 애플리케이션

- Node.JS용 노드 런타임 바이너리

지원하는 플랫폼

다음은 지원되는 플랫폼입니다(*는 임의 버전임).

Linux

- /usr/local/bin/node
- /usr/bin/node
- /nodejs/bin/node
- ~/.nvm/versions/node/*/bin/node
- ~/.local/share/fnm/node-versions/*/installation/bin/node
- ~/.asdf/installs/nodejs/*/bin/node
- ~/.local/share/mise/installs/node/*/bin/node
- ~/.volta/tools/image/node/*/bin/node

Windows

- C:\Program Files\nodejs\node.exe
- C:\Program Files (x86)\nodejs\node.exe
- ~\AppData\Roaming\fnm\node-versions*\installation\node.exe

macOS

- /opt/homebrew/Cellar/node/*/bin/node

주요 기능

이 컬렉션은 바이너리를 검사하여 임베디드 버전 정보를 추출합니다. 바이너리 실행 파일 `.rodata` 섹션(Linux의 ELF 바이너리), `.rdata` 섹션(Windows의 PE 바이너리) 또는 `__ctring` 섹션(MachO 바이너리)에서 버전 문자열을 검색합니다.

버전 문자열 예

다음은 Node.JS 런타임 바이너리에 포함된 버전 문자열의 예입니다.

```
node.js/v24.11.1
```

런타임 버전을 식별하기 위해 Node.JS 버전이 추출`24.11.1`됩니다.

예제 PURL

다음은 Node.JS의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/nodejs/node@24.11.1
```

OpenSSH 에코시스템 컬렉션

지원되는 애플리케이션

- OpenSSH (버전 9)
- OpenSSH (버전 10)

지원되는 플랫폼 Linux/macOS

- `/usr/sbin/sshd`
- `/usr/local/sbin/sshd`

지원되는 플랫폼 Windows

- `C:/Windows/System32/OpenSSH/sshd.exe`
- `C:/Program Files/OpenSSH/sshd.exe`
- `C:/Program Files (x86)/OpenSSH/sshd.exe`
- `C:/OpenSSH/sshd.exe`

주요 기능

- `sshd` 바이너리를 검사하여 임베디드 버전 정보를 추출합니다.
- 바이너리 실행 파일 `.rodata` 섹션(Linux의 ELF 바이너리, `__cstring` 섹션(MacOs의 Mach-O 바이너리) 또는 `.rdata` 섹션(Windows의 PE 바이너리)에서 버전 문자열을 찾습니다.

버전 문자열 예

다음은 OpenSSH 바이너리에 포함된 버전 문자열의 예입니다.

```
OpenSSH_9.9p2
```

OpenSSH 버전을 식별하기 위해 `9.9p2` 버전이 추출됩니다.

예제 PURL

다음은 OpenSSH의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/openssh/openssh@9.9p2
```

OpenSSL 에코시스템 컬렉션

지원되는 애플리케이션

OpenSSL 라이브러리 및 개발 패키지에 대한 지원은 3.0.0 이상 릴리스용 공식 OpenSSL로 빌드된 소프트웨어로 제한됩니다. 또한 소프트웨어는 시맨틱 버전 관리를 따라야 합니다. 사용자 지정 또는 포크된 OpenSSL 변형과 3.0.0 미만의 버전은 지원되지 않습니다.

Amazon Inspector SBOM 생성기는 설치된 각 OpenSSL 인스턴스에 대한 키 패키지 정보를 추출합니다.

주요 기능

- OpenSSL 헤더 파일에서 기본 SEMVER 버전 문자열을 추출합니다.
- OpenSSL 설치가 포함된 디렉터리 경로를 식별합니다.

Amazon Inspector SBOM 생성기는 플랫폼 전반의 공통 설치 경로에서 `openssl.h` 파일을 스캔하여 OpenSSL 설치를 찾습니다.

Linux/Unix 설치 경로 예제

다음은 Linux/Unix의 설치 경로 예제입니다.

```
/usr/local/include/openssl/openssl.h
/usr/local/ssl/include/openssl/openssl.h
/usr/local/openssl/include/openssl/openssl.h
/usr/local/opt/openssl/include/openssl/openssl.h
```

```
/usr/include/openssl/opensslv.h
```

Amazon Inspector SBOM 생성기는 `opensslv.h` 파일을 구문 분석하고 버전 정의를 찾아 버전 정보를 추출합니다.

```
# define OPENSSL_VERSION_MAJOR  3
# define OPENSSL_VERSION_MINOR  4
# define OPENSSL_VERSION_PATCH  0
```

예제 PURL

다음은 OpenSSL 버전의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/openssl/openssl@3.4.0
```

Oracle Database Server 컬렉션

지원되는 애플리케이션

- Oracle Database

지원되는 플랫폼 Linux

- `/opt/oracle`
- `/u01/app/oracle`

> **Note**
> 취약성 평가는 Oracle Database Server 버전 19 이상에만 적용됩니다.

주요 기능

- Oracle 바이너리를 검사하여 임베디드 버전 정보를 추출합니다.
- 바이너리 실행 파일 `.rodata` 섹션에서 버전 문자열을 찾습니다(Linux의 ELF 바이너리의 경우).
- 버전 정보는 RDBMS 버전 문자열을 포함하는 특정 형식을 따릅니다.

버전 문자열 예

다음은 Oracle Database 바이너리에 포함된 버전 문자열의 예입니다.

```
RDBMS_23.7.0.25.01DBRU_LINUX.X64_240304
```

Oracle Database 버전을 식별하기 위해 `23.7.0.25.01` 버전이 추출됩니다.

예제 PURL

다음은 Oracle Database의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/oracle/database@23.7.0.25.01
```

PHP 에코시스템 컬렉션

지원되는 애플리케이션

- PHP (버전 8.1 이상)

주요 기능

- 임베디드 버전 문자열을 사용하여 PHP 바이너리 실행 파일에서 버전 정보를 추출합니다.
- PHP 바이너리가 포함된 디렉터리 경로를 식별합니다.
- , 및와 같은 표준 PHP 바이너리 `php8.1` `php8.2`및 버전이 지정된 설치를 자동으로 감지합니다`php8.3`.

Amazon Inspector SBOM 생성기는 플랫폼 전반의 공통 PHP 설치 경로에서 설치를 찾습니다.

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` (버전이 지정된 바이너리)

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 (버전이 지정된 디렉터리)

PHP 버전 추출 예제

Amazon Inspector SBOM 생성기는 다음 패턴을 사용하여 임베디드 버전 문자열을 검색하여 PHP 바이너리에서 버전 정보를 추출합니다.

```
X-Powered-By: PHP/8.4.12
```

8.4.12는이 패턴에서 추출되어 PHP 버전을 식별합니다.

예제 PURL

다음은 PHP 패턴의 패키지 URL 예제입니다.

```
pkg:generic/php/php@8.4.12
```

WordPress 에코시스템 컬렉션

지원되는 구성 요소

- WordPress 코어
- WordPress 플러그인
- WordPress 테마

주요 기능

- WordPress 코어 - /wp-includes/version.php 파일을 구문 분석하여 $wp_version 변수에서 버전 값을 추출합니다.
- WordPress 플러그인 - /wp-content/plugins/<WordPress Plugin>/readme.txt 파일 또는 /wp-content/plugins/<WordPress Plugin>/readme.md 파일을 구문 분석하여 Stable 태그를 버전 문자열로 추출합니다.

- WordPress 테마 - `/wp-content/themes/<WordPress Theme>/style.css` 파일을 구문 분석하여 버전 메타데이터에서 버전을 추출합니다.

예시 **version.php** 파일

다음은 WordPress 코어 `version.php` 파일의 예제입니다.

```
// 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
```

예제 PURL

다음은 WordPress 코어의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/wordpress/core/wordpress@6.5.5
```

예시 **readme.txt** 파일

다음은 WordPress 플러그인 `readme.txt` 파일의 예제입니다.

```
=== 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
```

예제 PURL

다음은 WordPress 플러그인의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/wordpress/plugin/exclusive-addons-for-elementor@1.0.0
```

예시 **style.css** 파일

다음은 WordPress 테마 `style.css` 파일의 예입니다.

```
/*
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
*/
```

예제 PURL

다음은 WordPress 테마의 패키지 URL 예제입니다.

```
Sample PURL: pkg:generic/wordpress/theme/avada@1.0.0
```

Amazon Inspector SBOM 생성기 SSL/TLS 인증서 스캔

이 섹션에서는 Amazon Inspector SBOM 생성기를 사용하여 SSL/TLS 인증서를 인벤토리하는 방법을 설명합니다. Sbomgen은 사전 정의된 위치의 인증서와 사용자가 제공한 디렉터리를 검색하여 SSL/TLS 인증서를 인벤토리로 만듭니다. 이 특성은 사용자가 SSL/TLS 인증서를 인벤토리화하고 만료된 인증서를 식별할 수 있도록 하기 위한 것입니다. CA 인증서는 출력 인벤토리에도 표시됩니다.

Sbomgen 인증서 스캔 사용

`--scanners certificates` 인수를 사용하여 SSL/TLS 인증서 인벤토리 수집을 활성화할 수 있습니다. 인증서 스캔은 다른 스캐너와 결합할 수 있습니다. 기본적으로 인증서 스캔은 활성화되지 않습니다.

Sbomgen은 스캔 중인 아티팩트에 따라 여러 위치에서 인증서를 검색합니다. 모든 경우 Sbomgen이 다음 확장자가 있는 파일에서 인증서를 추출하려고 시도합니다.

```
.pem
.crt
.der
.p7b
.p7m
.p7s
.p12
.pfx
```

localhost 아티팩트 유형

인증서 스캐너가 활성화되어 있고 아티팩트 유형이 localhost인 경우 Sbomgen는 `*`가 비어 있지 않은 `/etc/*/ssl`, `/opt/*/ssl/certs`, `/usr/local/*/ssl` 및 `/var/lib/*/certs`에서 인증서를 반복적으로 찾습니다. 사용자가 제공한 디렉터리는 이름이 지정된 디렉터리에 관계없이 재귀적으

로 검색됩니다. 일반적으로 CA/시스템 인증서는 이러한 경로에 배치되지 않습니다. 이러한 인증서는 `pki`, `ca-certs` 또는 `CA` 폴더에 있는 경우가 많습니다. 기본 localhost 스캔 경로에도 나타날 수 있습니다.

디렉터리 및 컨테이너 아티팩트

디렉터리 또는 컨테이너 아티팩트를 스캔할 때는 Sbomgen에서 아티팩트의 모든 위치에 있는 인증서를 검색합니다.

예제 인증서 스캔 명령

다음은 예제 인증서 스캔 명령입니다. 하나는 로컬 디렉터리에 인증서만 포함하는 SBOM을 생성합니다. 또 다른 명령은 로컬 디렉터리에 인증서와 Alpine, Debian 및 Rhel 패키지가 포함된 SBOM을 생성합니다. 또 다른 명령은 공통 인증서 위치에 있는 인증서가 포함된 SBOM을 생성합니다.

```
# 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
```

예제 파일 구성 요소

다음은 인증서 조사 결과 구성 요소의 2가지 예입니다. 인증서가 만료되면 만료 날짜를 식별하는 추가 속성을 볼 수 있습니다.

```
{
      "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"
            }
        ]
}
```

취약성 대응 구성 요소의 예

`--scan-sbom` 플래그로 Amazon Inspector SBOM 생성기를 실행하면 취약성 스캔을 위해 결과 SBOM이 Amazon Inspector로 전송됩니다. 다음은 취약성 대응 구성 요소에 대한 인증서 조사 결과의 예입니다.

```
{
    "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"
}
```


Amazon Inspector SBOM 생성기 라이선스 컬렉션

Amazon Inspector SBOM 생성기는 소프트웨어 재료표(SBOM)의 라이선스 정보를 추적하는 데 도움이 됩니다. 운영 체제 및 프로그래밍 언어 전반에 걸쳐 지원되는 패키지에서 라이선스 정보를 수집합니다. 생성된 SBOM에서 표준화된 라이선스 표현식을 사용하면 라이선스 의무를 이해할 수 있습니다.

라이선스 정보 수집

명령 예제:

다음 예제에서는 디렉터리에서 라이선스 정보를 수집하는 방법을 보여줍니다.

```
./inspector-sbomgen directory --path /path/to/your/directory/ --collect-licenses
```

SBOM 구성 요소 예제

다음 예제에서는 생성된 SBOM의 구성 요소 항목을 보여줍니다.

```
"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",
    }
  ]
```

지원되는 패키지

라이선스 수집에는 다음 프로그래밍 언어 및 운영 체제 패키지가 지원됩니다.

대상	패키지 관리자	라이선스 정보 소스	Type
Alma Linux	RPM	• /usr/lib/sysimage/rpm/rpmdb.sqlite • /usr/lib/sysimage/rpm/Packages	OS

대상	패키지 관리자	라이선스 정보 소스	Type
		• /usr/lib/sysimage/rpm/Packages.db • /var/lib/rpm/rpmdb.sqlite • /var/lib/rpm/Packages • /var/lib/rpm/Packages.db	
Amazon Linux	RPM	• /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	OS

대상	패키지 관리자	라이선스 정보 소스	Type
CentOS	RPM	• /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	OS
Fedora	RPM	• /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	OS

대상	패키지 관리자	라이선스 정보 소스	Type
OpenSUSE	RPM	• /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	OS
Oracle Linux	RPM	• /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	OS

대상	패키지 관리자	라이선스 정보 소스	Type
Photon OS	RPM	• /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	OS
RHEL	RPM	• /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	OS

대상	패키지 관리자	라이선스 정보 소스	Type
Rocky Linux	RPM	• /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	OS
SLES	RPM	• /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	OS
Alpine Linux	APK	`/lib/apk/db/installed`	OS
Chainguard	APK	`/lib/apk/db/installed`	OS

대상	패키지 관리자	라이선스 정보 소스	Type
Debian	DPKG	`/usr/share/doc/*/copyright`	OS
Ubuntu	DPKG	`/usr/share/doc/*/copyright`	OS
Node.js	Javascript	`node_modules/*/package.json`	프로그래밍 언어
PHP	Composer 패키지	• `composer.lock` • `/vendor/composer/installed.json`	프로그래밍 언어
Go	Go	`LICENSE`	프로그래밍 언어
Python	Python/Egg/Wheel	• `.dist-info/METADATA` • `.egg-info` • `.egg-info/PKG-INFO`	프로그래밍 언어
Ruby	RubyGem	`*.gemspec`	프로그래밍 언어
Rust	crate	`Cargo.toml`	프로그래밍 언어

라이선스 표현식 표준화

SPDX 라이선스 표현식 형식은 오픈 소스 소프트웨어에 있는 라이선스 조건을 정확하게 표현합니다. Amazon Inspector SBOM 생성기는 이 섹션에 설명된 규칙을 통해 모든 라이선스 정보를 SPDX 라이선스 표현식으로 표준화합니다. 규칙은 라이선스 정보 전반의 일관성과 호환성을 제공합니다.

SPDX 짧은 양식 식별자 매핑

모든 라이선스 이름은 SPDX 짧은 양식 식별자에 매핑됩니다. 예를 들어 `MIT License`는 `MIT`로 축약됩니다.

다중 라이선스 조합

둘 이상의 라이선스를 `AND` 연산자와 결합할 수 있습니다. 다음 예제 명령에서는 명령의 형식을 지정하는 방법을 보여줍니다.

```
MIT AND Apache-2.0
```

사용자 지정 라이선스 접두사

사용자 지정 라이선스에는 `LicenseRef-CompanyPrivate`와 같이 `LicenseRef` 접두사가 붙습니다.

사용자 지정 예외 접두사

사용자 지정 예외에는 `AdditionRef-CustomException`와 같이 `AdditionRef-` 접두사가 붙습니다.

패키지 URL이란?

패키지 URL 또는 PURL은 다양한 패키지 관리 시스템에서 소프트웨어 패키지, 구성 요소 및 라이브러리를 식별하는 데 사용되는 표준화된 형식입니다. 이 형식을 사용하면 소프트웨어 프로젝트에서 종속성을 추적, 분석 및 관리하기가 더 쉬워지며, 특히 소프트웨어 재료표(SBOM)를 생성할 때 유용합니다.

PURL 구조

PURL 구조는 URL과 유사하며 여러 구성 요소로 구성됩니다.

- `pkg` - 리터럴 접두사
- `type` - 패키지 유형
- `namespace` - 그룹화
- `name` - 패키지 이름
- `version` - 패키지 버전
- `qualifiers` - 추가 키-값 페어
- `subpath` - 패키지의 파일 경로

예제 PURL

다음은 PURL이 어떻게 표시되는지 보여주는 예시입니다.

```
pkg:<type>/<namespace>/<name>@<version>?<qualifiers>#<subpath>
```

일반 PURL

일반 PURL은 npm, pypi 또는 maven과 같이 설정된 패키지 에코시스템에 맞지 않는 소프트웨어 패키지 및 구성 요소를 나타내는 데 사용됩니다. 소프트웨어 구성 요소를 식별하고 특정 패키지 관리 시스템과 일치하지 않을 수 있는 메타데이터를 캡처합니다. 일반 PURL은 컴파일된 바이너리부터 Apache 및 WordPress와 같은 플랫폼에 이르기까지 다양한 소프트웨어 프로젝트에 유용합니다. 이를 통해 컴파일된 바이너리, 웹 플랫폼, 사용자 지정 소프트웨어 배포 등 다양한 사용 사례에 적용할 수 있습니다.

주요 사용 사례

- 컴파일된 바이너리를 지원하며 Go 및 Rust에 유용합니다.
- 패키지가 기존 패키지 관리자와 연결되지 않을 수 있는 Apache 및 WordPress와 같은 웹 플랫폼을 지원합니다.
- 조직이 내부적으로 개발된 소프트웨어 또는 공식 패키지가 없는 시스템을 참조할 수 있도록 하여 사용자 지정 레거시 소프트웨어를 지원합니다.

예제 형식

다음은 일반 PURL 형식의 예제입니다.

```
pkg:generic/<namespace>/<name>@<version>?<qualifiers>
```

일반 PURL 형식의 추가 예제

다음은 일반 PURL 형식의 추가 예제입니다.

컴파일된 Go 바이너리

다음은 Go로 컴파일된 `inspector-sbomgen binary`를 나타냅니다.

```
pkg:generic/inspector-sbomgen?go_toolchain=1.22.5
```

컴파일된 Rust 바이너리

다음은 Rust로 컴파일된 myrustapp 바이너리를 나타냅니다.

```
pkg:generic/myrustapp?rust_toolchain=1.71.0
```

Apache 프로젝트

다음은 Apache 네임스페이스 아래의 http 프로젝트를 나타냅니다.

```
pkg:generic/apache/httpd@1.0.0
```

WordPress 소프트웨어

다음은 코어 WordPress 소프트웨어를 나타냅니다.

```
pkg:generic/wordpress/core/wordpress@6.0.0
```

WordPress 테마

다음은 사용자 지정 WordPress 테마를 나타냅니다.

```
pkg:generic/wordpress/theme/mytheme@1.0.0
```

WordPress 플러그인

다음은 사용자 지정 WordPress 플러그인을 나타냅니다.

```
pkg:generic/wordpress/plugin/myplugin@1.0.0
```

Amazon Inspector SBOM 생성기에서 해결되지 않은 버전 참조 또는 비표준 버전 참조 처리

Amazon Inspector SBOM 생성기는 소스 파일에서 직접 종속성을 식별하여 시스템 내에서 지원되는 아티팩트를 찾고 구문 분석합니다. 패키지 관리자가 아니며 버전 범위를 확인하거나, 동적 참조를 기반으로 버전을 추론하거나, 레지스트리 조회를 처리하지 않습니다. 프로젝트 소스 아티팩트에 정의된 대로만 종속성을 수집합니다. 대부분의 경우, `package.json`, `pom.xml` 또는 `requirements.txt`와 같은 패키지 매니페스트의 종속성은 해결되지 않은 버전 또는 범위 기반 버전을 사용하여 지정됩니다. 이 주제에는 이러한 종속성이 어떻게 보일 수 있는지에 대한 예제가 포함되어 있습니다.

권장 사항

Amazon Inspector SBOM 생성기는 소스 아티팩트에서 종속성을 추출하지만 버전 범위 또는 동적 참조를 해석하지는 않습니다. 취약성 스캔 및 SBOM 정확도를 높이려면 프로젝트 종속성에서 해결된 의미 체계 버전 식별자를 사용하는 것이 좋습니다.

Java

Java의 경우 Maven 프로젝트는 버전 범위를 사용하여 `pom.xml` 파일에서 종속성을 정의할 수 있습니다.

```
<dependency>
    <groupId>org.inspector</groupId>
    <artifactId>inspector-api</artifactId>
    <version>(,1.0]</version>
</dependency>
```

범위는 1.0 이하의 모든 버전이 허용되도록 지정합니다. 그러나 버전이 확인된 버전이 아닌 경우 Amazon Inspector SBOM 생성기는 특정 릴리스에 매핑할 수 없으므로 해당 버전을 수집하지 않습니다.

JavaScript

JavaScript의 경우 `package.json` 파일에는 다음과 유사한 버전 범위가 포함될 수 있습니다.

```
"dependencies": {
    "ky": "^1.2.0",
    "registry-auth-token": "^5.0.2",
    "registry-url": "^6.0.1",
    "semver": "^7.6.0"
}
```

^ 연산자는 지정된 버전보다 크거나 같은 버전이 허용되도록 지정합니다. 그러나 지정된 버전이 확인된 버전이 아닌 경우 Amazon Inspector SBOM 생성기는 이를 수집하지 않습니다. 이렇게 하면 취약성 감지 중에 오탐지가 발생할 수 있습니다.

Python

Python의 경우 `requirements.txt` 파일에 부울 표현식이 있는 항목이 포함될 수 있습니다.

```
requests>=1.0.0
```

>= 연산자는 `1.0.0`보다 크거나 같은 버전이 허용되도록 지정합니다. 이 특정 표현식은 정확한 버전을 지정하지 않으므로 Amazon Inspector SBOM 생성기는 취약성 분석을 위해 버전을 안정적으로 수집할 수 없습니다.

Amazon Inspector SBOM 생성기는 베타, 최신 또는 스냅샷과 같은 비표준 또는 모호한 버전 식별자를 지원하지 않습니다.

```
pkg:maven/org.example.com/testmaven@1.0.2%20Beta-RC-1_Release
```

> **Note**
>
> Beta-RC-1_Release와 같은 비표준 접미사 사용은 표준 의미 체계 버전 관리를 준수하지 않으며 Amazon Inspector 감지 엔진 내의 취약성을 평가할 수 없습니다.

Amazon Inspector에서 CycloneDX 네임스페이스 사용

Amazon Inspector는 SBOM과 함께 사용할 수 있는 CycloneDX 네임스페이스 및 속성 이름을 제공합니다. 이 단원에서는 CycloneDX SBOM의 구성 요소에 추가될 수 있는 모든 사용자 지정 키/값 속성에 대해 설명합니다. 자세한 내용은 GitHub 웹 사이트에서 CycloneDX property taxonomy를 참조하세요.

`amazon:inspector:sbom_scanner` 네임스페이스 분류법

Amazon Inspector 스캔 API는 `amazon:inspector:sbom_scanner` 네임스페이스를 사용하며 다음과 같은 속성을 제공합니다.

속성	설명
`amazon:inspector:sbom_scanner:cisa_kev_date_added`	취약성이 CISA의 알려진 악용된 취약성 카탈로그에 추가된 시기를 나타냅니다.

속성	설명
amazon:inspector:sbom_scanner:cisa_kev_date_due	CISA의 알려진 악용된 취약성 카탈로그에 따라 취약성 수정이 마감되는 시기를 나타냅니다.
amazon:inspector:sbom_scanner:critical_vulnerabilities	SBOM에서 발견된 심각한 심각도 취약성의 총 개수.
amazon:inspector:sbom_scanner:exploit_available	해당 취약성에 대한 악용이 가능한지를 나타냅니다.
amazon:inspector:sbom_scanner:exploit_last_seen_in_public	해당 취약성에 대한 악용이 공개적으로 마지막으로 발견된 시기를 나타냅니다.
amazon:inspector:sbom_scanner:fixed_version: *component_bom_ref*	지정된 취약성에 대해 표시된 구성 요소의 수정된 버전을 제공합니다.
amazon:inspector:sbom_scanner:high_vulnerabilities	SBOM에서 발견된 높은 심각도 취약성의 총 개수.
amazon:inspector:sbom_scanner:info	특정 구성 요소에 대한 스캔 컨텍스트를 제공합니다(예: '구성 요소 검사됨: 발견된 취약성 없음').
amazon:inspector:sbom_scanner:is_malicious	OpenSSF가 영향을 받는 구성 요소를 악성으로 식별하는지 여부를 나타냅니다.
amazon:inspector:sbom_scanner:low_vulnerabilities	SBOM에서 발견된 낮은 심각도 취약성의 총 개수.
amazon:inspector:sbom_scanner:medium_vulnerabilities	SBOM에서 발견된 중간 심각도 취약성의 총 개수.
amazon:inspector:sbom_scanner:path	대상 패키지 정보를 생성하는 파일의 경로입니다.

속성	설명
amazon:inspector:sbom_scanner:priority	지정된 취약성을 수정하는 데 권장되는 우선 순위입니다. 값은 내림차순으로 'IMMEDIATE', 'URGENT', 'MODERATE', 'STANDARD'입니다.
amazon:inspector:sbom_scanner:priority_intelligence	지정된 취약성의 우선순위를 결정하는 데 사용되는 인텔리전스의 품질입니다. 값은 'VERIFIED' 또는 'UNVERIFIED'입니다.
amazon:inspector:sbom_scanner:warning	특정 구성 요소가 스캔되지 않은 이유에 대한 컨텍스트를 제공합니다(예: '구성 요소 건너뜀: purl 제공 안 됨').

amazon:inspector:sbom_generator 네임스페이스 분류법

Amazon Inspector SBOM 생성기는 amazon:inspector:sbom_generator 네임스페이스를 사용하며 다음과 같은 속성을 제공합니다.

속성	설명
amazon:inspector:sbom_generator:cpu_architecture	인벤토리 대상 시스템의 CPU 아키텍처(x86_64).
amazon:inspector:sbom_generator:ec2:instance_id	Amazon EC2 인스턴스 ID입니다.
amazon:inspector:sbom_generator:ec2:instance_type	Amazon EC2 인스턴스 유형
amazon:inspector:sbom_generator:live_patching_enabled	Amazon EC2 Amazon Linux에서 라이브 패치가 활성화되었는지 여부를 나타내는 부울 값입니다.
amazon:inspector:sbom_generator:live_patched_cves	Amazon EC2 Amazon Linux에서 라이브 패치를 통해 패치된 CVE 목록입니다.

속성	설명
amazon:inspector:sbom_generator:dockerfile_finding: *inspector_finding_id*	구성 요소의 Amazon Inspector 조사 결과가 Dockerfile 검사와 관련이 있음을 나타냅니다.
amazon:inspector:sbom_generator:image_id	컨테이너 이미지 구성 파일에 속하는 해시입니다(이미지 ID라고도 함).
amazon:inspector:sbom_generator:image_arch	컨테이너 이미지의 아키텍처입니다.
amazon:inspector:sbom_generator:image_author	컨테이너 이미지의 작성자입니다.
amazon:inspector:sbom_generator:image_docker_version	컨테이너 이미지를 빌드하는 데 사용되는 도커 버전입니다.
amazon:inspector:sbom_generator:is_duplicate_package	두 개 이상의 파일 스캐너에서 주제 패키지를 찾았음을 나타냅니다.
amazon:inspector:sbom_generator:duplicate_purl	다른 스캐너에서 찾은 중복 패키지 PURL을 나타냅니다.
amazon:inspector:sbom_generator:kernel_name	인벤토리 대상 시스템의 커널 이름.
amazon:inspector:sbom_generator:kernel_version	인벤토리 대상 시스템의 커널 버전.
amazon:inspector:sbom_generator:kernel_component	주제 패키지가 커널 구성 요소인지 여부를 나타내는 부울 값입니다.
amazon:inspector:sbom_generator:running_kernel	제목 패키지가 실행 중인 커널인지 여부를 나타내는 부울 값입니다.
amazon:inspector:sbom_generator:layer_diff_id	압축되지 않은 컨테이너 이미지 계층의 해시.

속성	설명
amazon:inspector:sbom_generator:replaced_by	현재 Go 모듈을 대체하는 값입니다.
amazon:inspector:sbom_generator:os_hostname	인벤토리 대상 시스템의 호스트 이름.
amazon:inspector:sbom_generator:source_file_scanner	패키지 정보가 들어 있는 파일을 찾은 스캐너(예: /var/lib/dpkg/status).
amazon:inspector:sbom_generator:source_package_collector	특정 파일에서 패키지 이름과 버전을 추출한 컬렉터.
amazon:inspector:sbom_generator:source_path	주제 패키지 정보가 추출된 파일의 경로.
amazon:inspector:sbom_generator:file_size_bytes	지정된 아티팩트의 파일 크기를 나타냅니다.
amazon:inspector:sbom_generator:unresolved_version	패키지 관리자가 확인하지 않은 버전 문자열을 나타냅니다.
amazon:inspector:sbom_generator:experimental:transitive_dependency	패키지 관리자의 간접 종속성을 나타냅니다.
amazon:inspector:sbom_generator:metadata:host:hostname	스캔한 시스템의 호스트 이름입니다.
amazon:inspector:sbom_generator:metadata:host:kernel_name	운영 체제의 커널 이름(예: Linux, Darwin, Windows_NT).
amazon:inspector:sbom_generator:metadata:host:kernel_version	운영 체제의 커널 버전 문자열입니다.

속성	설명
amazon:inspector:sbom_generator:metadata:host:cpu_architecture	시스템의 CPU 아키텍처(예: x86_64, arm64).
amazon:inspector:sbom_generator:metadata:host:bootdisk_id	부팅 디스크의 고유 식별자입니다.
amazon:inspector:sbom_generator:metadata:host:boot_id	현재 부팅 세션의 고유 식별자입니다.
amazon:inspector:sbom_generator:metadata:host:boot_time	ISO 8601 형식의 시스템 부팅 시간입니다.
amazon:inspector:sbom_generator:metadata:host:system_id	영구 시스템 식별자(Linux의 Machine-id, Windows의 MachineGuid).
amazon:inspector:sbom_generator:metadata:host:system_serial	시스템 펌웨어의 하드웨어 일련 번호입니다.
amazon:inspector:sbom_generator:metadata:host:network_interfaces: *name*:hardware	네트워크 인터페이스의 MAC 주소입니다.
amazon:inspector:sbom_generator:metadata:host:network_interfaces: *name*:ipv4	인터페이스에 할당된 IPv4 주소(들)입니다.
amazon:inspector:sbom_generator:metadata:host:network_interfaces: *name*:ipv6	인터페이스에 할당된 IPv6 주소(들)입니다.
amazon:inspector:sbom_generator:metadata:host:sbomgen_tag: *key*	--tag CLI 인수를 통해 전달되는 사용자 정의 태그입니다.

속성	설명
amazon:inspector:sbom_generator:metadata:imds:provider	IMDS(aws, azure)를 통해 감지된 클라우드 공급자입니다.
amazon:inspector:sbom_generator:metadata:imds:instance_id	Amazon EC2 인스턴스 ID 또는 Azure VM 이름입니다.
amazon:inspector:sbom_generator:metadata:imds:instance_type	인스턴스 유형(예: t3.micro, Standard_D2s_v3).
amazon:inspector:sbom_generator:metadata:imds:instance_location	인스턴스의 리전/위치입니다.
amazon:inspector:sbom_generator:metadata:imds:instance_partition	클라우드 파티션(aws, aws-cn, aws-us-gov AWS for 또는 AzurePublicCloud for Azure).
amazon:inspector:sbom_generator:metadata:imds:instance_managed_id	Amazon EC2 Systems Manager 관리형 인스턴스 ID(AWS 만 해당).
amazon:inspector:sbom_generator:metadata:imds:tenant_id	Azure 테넌트 ID(Azure만 해당).
amazon:inspector:sbom_generator:metadata:imds:vm_id	Azure VM 고유 식별자(Azure만 해당).
amazon:inspector:sbom_generator:metadata:host:open_port: *port*:*protocol*	런타임 리소스(예: EC2)의 열린 포트를 나타냅니다.
amazon:inspector:sbom_generator:hardened_image:vendor	강화된 컨테이너 이미지의 공급업체

Amazon Inspector 스캔을 CI/CD 파이프라인에 통합

Amazon Inspector CI/CD 통합은 Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 활용하여 컨테이너 이미지에 대한 취약성 보고서를 생성합니다. Amazon Inspector SBOM 생성기는 아카이브, 컨테이너 이미지, 디렉터리, 로컬 시스템, 컴파일된 Go 및 Rust 바이너리에 대해 소프트웨어 자재 명세서(SBOM)를 생성합니다. Amazon Inspector 스캔 API는 SBOM을 스캔하여 탐지된 취약성에 대한 세부 정보가 포함된 보고서를 생성합니다. Amazon Inspector 컨테이너 이미지 스캔을 CI/CD 파이프라인에 통합하여 소프트웨어 취약성을 스캔하고 취약성 보고서를 생성하여 배포 전에 위험을 조사하고 해결할 수 있습니다. CI/CD 통합을 설정하려면 플러그인을 사용하거나, Amazon Inspector SBOM 생성기 및 Amazon Inspector 스캔 API를 사용하여 사용자 지정 CI/CD 통합을 생성할 수 있습니다.

주제

- 플러그인 통합
- 사용자 지정 통합
- Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정
- Amazon Inspector Dockerfile 검사
- Amazon Inspector 스캔을 사용하여 사용자 지정 CI/CD 파이프라인 통합 생성
- Amazon Inspector Jenkins 플러그인 사용
- Amazon Inspector TeamCity 플러그인 사용
- GitHub 작업과 함께 Amazon Inspector 사용
- GitLab 구성 요소와 함께 Amazon Inspector 사용
- Amazon Inspector와 함께 CodeCatalyst 작업 사용
- CodePipeline에서 Amazon Inspector 스캔 작업 사용

플러그인 통합

Amazon Inspector는 지원되는 CI/CD 솔루션을 위한 플러그인을 제공합니다. 각 마켓플레이스에서 이러한 플러그인을 설치한 다음 이를 사용하여 Amazon Inspector 스캔을 파이프라인의 빌드 단계로 추가할 수 있습니다. 플러그인 빌드 단계에서는 제공한 이미지에서 Amazon Inspector SBOM 생성기를 실행한 다음, 생성된 SBOM에서 Amazon Inspector 스캔 API를 실행합니다.

다음은 Amazon Inspector CI/CD 통합이 플러그인을 통해 작동하는 방식에 대한 개요입니다.

1. Amazon Inspector 스캔 API에 대한 액세스를 허용 AWS 계정 하도록를 구성합니다. 지침은 Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정 섹션을 참조하세요.
2. 마켓플레이스에서 Amazon Inspector 플러그인을 설치합니다.
3. Amazon Inspector SBOM 생성기 바이너리를 설치하고 구성합니다. 지침은 Amazon Inspector SBOM 생성기 섹션을 참조하세요.
4. Amazon Inspector 스캔을 CI/CD 파이프라인의 빌드 단계로 추가하고 스캔을 구성합니다.
5. 빌드를 실행하면 플러그인이 컨테이너 이미지를 입력으로 받은 다음 이미지에서 Amazon Inspector SBOM 생성기를 실행하여 CycloneDX와 호환되는 SBOM을 생성합니다.
6. 그러면 플러그인은 생성된 SBOM을 Amazon Inspector 스캔 API 엔드포인트로 전송합니다. Amazon Inspector 스캔 API 엔드포인트는 각 SBOM 구성 요소의 취약성을 평가합니다.
7. Amazon Inspector 스캔 API 응답은 CSV, SBOM, JSON 및 HTML 형식의 취약성 보고서로 변환됩니다. 보고서에는 Amazon Inspector에서 발견한 모든 취약성에 대한 세부 정보가 포함되어 있습니다.

지원되는 CI/CD 솔루션

Amazon Inspector는 현재 다음과 같은 CI/CD 솔루션을 지원합니다. 플러그인을 사용하여 CI/CD 통합을 설정하는 방법에 대한 전체 지침을 보려면 CI/CD 솔루션용 플러그인을 선택하세요.

- Jenkins 플러그인
- TeamCity 플러그인
- GitHub 작업

사용자 지정 통합

Amazon Inspector에서 CI/CD 솔루션용 플러그인을 제공하지 않는 경우, Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 함께 사용하여 사용자 지정 CI/CD 통합을 생성할 수 있습니다. 또한 사용자 지정 통합을 사용하여 Amazon Inspector SBOM 생성기를 통해 제공되는 옵션을 사용하여 스캔을 미세 조정할 수 있습니다.

다음은 사용자 지정 Amazon Inspector CI/CD 통합이 작동하는 방식에 대한 개요입니다.

1. Amazon Inspector 스캔 API에 대한 액세스를 허용 AWS 계정 하도록를 구성합니다. 지침은 Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정 섹션을 참조하세요.

2. Amazon Inspector SBOM 생성기 바이너리를 설치하고 구성합니다. 지침은 Amazon Inspector SBOM 생성기 섹션을 참조하세요.
3. Amazon Inspector SBOM 생성기를 사용하여 컨테이너 이미지에 대해 CycloneDX와 호환되는 SBOM을 생성합니다.
4. 생성된 SBOM에서 Amazon Inspector 스캔 API를 사용하여 취약성 보고서를 생성합니다.

사용자 지정 통합을 설정하는 방법에 대한 지침은 Amazon Inspector 스캔을 사용하여 사용자 지정 CI/CD 파이프라인 통합 생성 섹션을 참조하세요.

Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정

Amazon Inspector CI/CD 통합을 사용하려면 AWS 계정에 가입해야 합니다. 에는 CI/CD pipleline에 Amazon Inspector Scan API에 대한 액세스 권한을 부여하는 IAM 역할이 있어야 AWS 계정 합니다. 다음 주제의 작업을 완료하여에 가입하고 AWS 계정, 관리자 사용자를 생성하고, CI/CD 통합을 위한 IAM 역할을 구성합니다.

> **Note**
>
> 에 이미 가입한 경우 로 건너뛸 AWS 계정수 있습니다CI/CD 통합을 위한 IAM 역할 구성.

주제

- 에 가입 AWS 계정
- 관리자 액세스 권한이 있는 사용자 생성
- CI/CD 통합을 위한 IAM 역할 구성

에 가입 AWS 계정

이 없는 경우 다음 단계를 AWS 계정완료하여 생성합니다.

에 가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따르세요.

   등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

에 가입하면 AWS 계정AWS 계정 루트 사용자이 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/으로 이동하고 내 계정을 선택하여 현재 계정 활동을 확인하고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 AWS 계정보호 AWS IAM Identity Center, AWS 계정 루트 사용자활성화 및 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자AWS Management Console로에 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하세요.
2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자(콘솔)에 대한 가상 MFA 디바이스 활성화를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 자격 증명 소스 IAM Identity Center 디렉터리 로 사용하는 방법에 대한 자습서는 사용 AWS IAM Identity Center 설명서의 기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요. AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.
2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 그룹 추가를 참조하세요.

CI/CD 통합을 위한 IAM 역할 구성

Amazon Inspector 스캔을 CI/CD 파이프라인에 통합하려면 소프트웨어 재료 명세서(SBOM)를 스캔하는 Amazon Inspector 스캔 API에 대한 액세스를 허용하는 IAM 정책을 생성해야 합니다. 그런 다음 Amazon Inspector 스캔 API를 실행하기 위해 계정에서 위임할 수 있는 IAM 역할에 해당 정책을 연결할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
3. 정책 편집기에서 JSON을 선택한 후 다음 문을 붙여 넣습니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
```

```
            "Action": "inspector-scan:ScanSbom",
            "Resource": "*"
        }
    ]
}
```

4. 다음을 선택합니다.
5. 정책 이름(예: `InspectorCICDscan-policy`)과 선택 사항인 설명을 입력한 다음 정책 생성을 선택합니다. 이 정책은 다음 단계에서 생성할 역할에 연결됩니다.
6. IAM 콘솔의 탐색 창에서 역할을 선택한 다음 새 역할 생성을 선택합니다.
7. 신뢰할 수 있는 엔터티 유형에서 사용자 지정 신뢰 정책을 선택한 후 다음 정책을 입력합니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

8. 다음을 선택합니다.
9. 권한 추가에서 앞서 생성한 정책을 검색하여 선택하고 다음을 선택합니다.
10. 역할 이름(예: `InspectorCICDscan-role`)과 선택 사항인 설명을 입력한 다음 `Create Role`을 선택합니다.

Amazon Inspector Dockerfile 검사

이 단원에서는 Amazon Inspector SBOM 생성기를 사용하여 보안 취약성을 유발하는 잘못된 구성이 있는지 Dockerfiles 및 Docker 컨테이너 이미지를 스캔하는 방법에 대해 설명합니다.

주제

- Sbomgen Dockerfile 검사 사용
- 지원되는 Dockerfile 검사

Sbomgen Dockerfile 검사 사용

Dockerfile 검사는 `Dockerfile` 또는 `*.Dockerfile`이라는 파일이 검색되고 Docker 이미지가 스캔될 때 자동으로 수행됩니다.

`--skip-scanners dockerfile` 인수를 사용하여 Dockerfile 검사를 비활성화할 수 있습니다. 또한 Dockerfile 검사를 OS 또는 타사 패키지 등 사용 가능한 모든 스캐너와 결합할 수도 있습니다.

Docker 검사 명령 예제

다음 예제 명령은 Dockerfile과 Docker 컨테이너 이미지, OS 및 타사 패키지에 대한 SBOM을 생성하는 방법을 보여줍니다.

```
# 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
```

예제 파일 구성 요소

다음은 파일 구성 요소에 대한 Dockerfile 조사 결과의 예입니다.

```
{
     "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"
    },
```

취약성 대응 구성 요소의 예

다음은 취약성 대응 구성 요소에 대한 Dockerfile 조사 결과의 예입니다.

```
{
      "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**
>
> Sbomgen을 `--scan-sbom` 플래그 없이 간접적으로 호출하는 경우 원시 Dockerfile 조사 결과만 볼 수 있습니다.

지원되는 Dockerfile 검사

Sbomgen Dockerfile 검사는 다음 항목에 대해 지원됩니다.

- Sudo 바이너리 패키지
- Debian APT 유틸리티
- 하드코딩된 보안 암호
- 루트 컨테이너
- 런타임 약화 명령 플래그
- 런타임 약화 환경 변수

이러한 각 Dockerfile 검사의 심각도 등급은 다음 주제의 상단에 나와 있습니다.

> **Note**
>
> 다음 주제에 설명된 권장 사항은 업계 모범 사례를 기반으로 합니다.

Sudo 바이너리 패키지

> **Note**
>
> 이 검사의 심각도 등급은 정보입니다.

Sudo 바이너리 패키지는 예측할 수 없는 TTY 및 신호 전달 동작이 발생할 수 있으므로 설치하거나 사용하지 않는 것이 좋습니다. 자세한 내용은 Docker Docs 웹 사이트에서 User를 참조하세요. 사용 사례에 Sudo 바이너리 패키지와 유사한 기능이 필요한 경우 Gosu를 사용하는 것이 좋습니다.

Debian APT 유틸리티

> **Note**
>
> 이 검사의 심각도 등급은 높음입니다.

다음은 Debian APT 유틸리티 사용에 대한 모범 사례입니다.

캐싱 문제 방지를 위해 단일 **Run** 문에 **apt-get** 명령 결합

Docker 컨테이너 내부의 단일 RUN 문에 `apt-get` 명령을 결합하는 것이 좋습니다. `apt-get update`만 단독으로 사용하면 캐싱 문제가 발생하고 후속 `apt-get install` 명령이 실패할 수 있습니다. 자세한 내용은 Docker Docs 웹 사이트에서 apt-get을 참조하세요.

> **Note**
>
> 설명한 캐싱 동작은 Docker 컨테이너 소프트웨어가 최신 버전이 아닌 경우에도 Docker 컨테이너 내부에서 발생할 수 있습니다.

비대화형 방식으로 APT 명령줄 유틸리티 사용

APT 명령줄 유틸리티는 대화형으로 사용하는 것이 좋습니다. APT 명령줄 유틸리티는 최종 사용자 도구로 설계되었으며 버전에 따라 동작이 달라집니다. 자세한 내용은 Debian 웹 사이트에서 Script Usage and differences from other APT Tools를 참조하세요.

하드 코딩된 보안 암호

> **Note**
>
> 이 검사의 심각도 등급은 중요입니다.

Dockerfile의 기밀 정보는 하드 코딩된 보안 암호로 간주됩니다. Sbomgen Docker 파일 검사를 통해 다음과 같은 하드 코딩된 보안 암호를 식별할 수 있습니다.

- AWS 액세스 키 IDs- `AKIAIOSFODNN7EXAMPLE`
- AWS 보안 키 - `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

- DockerHub 개인 액세스 토큰 - `dckr_pat_thisisa27charexample1234567`
- GitHub 개인 액세스 토큰 - `ghp_examplev61wY7Pj1YnotrealUoY123456789`
- GitLab 개인 액세스 토큰 - `glpat-12345example12345678`

루트 컨테이너

> **Note**
>
> 이 검사의 심각도 마커는 정보입니다.

Docker 컨테이너는 루트 권한 없이 실행하는 것이 좋습니다. 루트 권한 없이 실행할 수 없는 컨테이너화된 워크로드의 경우 최소 권한 원칙을 기반으로 애플리케이션을 구축하는 것이 좋습니다. 자세한 내용은 Docker Docs 웹 사이트에서 User를 참조하세요.

런타임 약화 환경 변수

> **Note**
>
> 이 검사의 심각도 등급은 높음입니다.

여러 명령줄 유틸리티 또는 프로그래밍 언어 런타임은 보안 기본값 우회를 지원하므로 안전하지 않은 방법을 통해 실행될 수 있습니다.

NODE_TLS_REJECT_UNAUTHORIZED=0

Node.js 프로세스가 `NODE_TLS_REJECT_UNAUTHORIZED`를 `0`으로 설정하여 실행되면 TLS 인증서 검증이 비활성화됩니다. 자세한 내용은 Node.js 웹 사이트에서 NODE_TLS_REJECT_UNAUTHORIZED=0을 참조하세요.

GIT_SSL_NO_VERIFY=*

`GIT_SSL_NO_VERIFY`를 설정한 상태에서 git 명령줄 프로세스를 실행하면 Git은 TLS 인증서 확인을 건너뜁니다. 자세한 내용은 Git 웹 사이트에서 Environment variables를 참조하세요.

PIP_TRUSTED_HOST=*

`PIP_TRUSTED_HOST`를 설정한 상태에서 Python pip 명령을 실행하면 Pip은 지정된 도메인에서 TLS 인증서 확인을 건너뜁니다. 자세한 내용은 Pip 웹 사이트에서 --trusted-host를 참조하세요.

NPM_CONFIG_STRICT_SSL=false

Node.js npm 명령줄 프로세스가 NPM_CONFIG_STRICT_SSL을 false로 설정하여 실행되면 Node Package Manager(npm) 유틸리티는 TLS 인증서를 검증하지 않고 NPM 레지스트리에 연결합니다. 자세한 내용은 npm Docs 웹 사이트에서 strict-ssl을 참조하세요.

런타임 약화 명령 플래그

> **Note**
>
> 이 검사의 심각도 등급은 높음입니다.

런타임 약화 환경 변수와 마찬가지로, 여러 명령줄 유틸리티 또는 프로그래밍 언어 런타임은 보안 기본값 우회를 지원하므로 안전하지 않은 방법을 통해 실행될 수 있습니다.

npm --strict-ssl=false

Node.js npm 명령줄 프로세스가 --strict-ssl=false 플래그와 함께 실행되면 Node Package Manager(npm) 유틸리티는 TLS 인증서를 검증하지 않고 NPM 레지스트리에 연결됩니다. 자세한 내용은 npm Docs 웹 사이트에서 strict-ssl을 참조하세요.

apk --allow-untrusted

Alpine Package Keeper 유틸리티가 --allow-untrusted 플래그와 함께 실행되면 apk는 서명이 없거나 신뢰할 수 없는 패키지를 설치합니다. 자세한 내용은 Apline 웹 사이트에서 다음 리포지토리를 참조하세요.

apt-get --allow-unauthenticated

Debian apt-get 패키지 유틸리티를 --allow-unauthenticated 플래그와 함께 실행하면 apt-get은 패키지의 유효성을 검사하지 않습니다. 자세한 내용은 Debian 웹 사이트에서 APT-Get(8)을 참조하세요.

pip --trusted-host

Python pip 유틸리티를 --trusted-host 플래그와 함께 실행하면 지정된 호스트 이름이 TLS 인증서 검증을 건너뜁니다. 자세한 내용은 Pip 웹 사이트에서 --trusted-host를 참조하세요.

rpm --nodigest, --nosignature, --noverify, --nofiledigest

RPM 기반 패키지 관리자 `rpm`이 `--nodigest`, `--nosignature`, `--noverify`, `--nofiledigest` 플래그와 함께 실행되면 RPM 패키지 관리자는 패키지를 설치할 때 패키지 헤더, 서명 또는 파일의 유효성을 검사하지 않습니다. 자세한 내용은 RPM 웹 사이트에서 다음 RPM 매뉴얼 페이지를 참조하세요.

`yum-config-manager --setopt=sslverify false`

RPM 기반 패키지 관리자 `yum-config-manager`가 `--setopt=sslverify` 플래그를 false로 설정하여 실행되면 YUM 패키지 관리자는 TLS 인증서를 검증하지 않습니다. 자세한 내용은 Man7 웹 사이트에서 다음 YUM 매뉴얼 페이지를 참조하세요.

`yum --nogpgcheck`

RPM 기반 패키지 관리자 `yum`이 `--nogpgcheck` 플래그와 함께 실행되면 YUM 패키지 관리자는 패키지의 GPG 서명 검사를 건너뜁니다. 자세한 내용은 Man7 웹 사이트에서 yum(8)을 참조하세요.

`curl --insecure, curl -k`

`curl`이 `--insecure` 또는 `-k` 플래그와 함께 실행되면 TLS 인증서 검증이 비활성화됩니다. 기본적으로 `curl`에서 설정하는 모든 보안 연결은 전송이 발생하기 전에 보안 상태인지 확인을 거칩니다. 이 옵션을 사용하면 `curl`은 확인 단계를 건너뛰고 검사 없이 진행합니다. 자세한 내용은 Curl 웹 사이트에서 다음 Curl 매뉴얼 페이지를 참조하세요.

`wget --no-check-certificate`

`wget`이 `--no-check-certificate` 플래그와 함께 실행되면 TLS 인증서 검증이 비활성화됩니다. 자세한 내용은 GNU 웹 사이트에서 다음 Wget 매뉴얼 페이지를 참조하세요.

컨테이너 내 OS 패키지 데이터베이스 제거 확인

> **Note**
>
> 이 검사의 심각도 등급은 정보입니다.

운영 체제 패키지 데이터베이스를 제거하면 컨테이너 이미지 소프트웨어의 전체 인벤토리를 스캔하는 기능이 줄어듭니다. 이러한 데이터베이스는 컨테이너 빌드 단계에서 그대로 두어야 합니다.

OS 패키지 데이터베이스에 대한 제거 검사는 다음 패키지 관리자에 대해 지원됩니다.

Alpine 패키지 키퍼(APK)

설치된 소프트웨어에 대해 APK 패키지 관리자를 사용하는 컨테이너 이미지는 빌드 중에 APK 시스템 파일이 제거되지 않도록 해야 합니다. 자세한 내용은 Arch Linux 웹 사이트의 APK 관리 시스템 파일 설명서를 참조하세요.

Debian Package Manager(DPKG)

Debian, Ubuntu 또는 Distroless 기반 이미지와 같이 DPKG 패키지 관리자를 사용하는 컨테이너는 컨테이너 빌드 중에 DPKG 데이터베이스가 제거되지 않았는지 확인해야 합니다. 자세한 내용은 Ubuntu 웹 사이트의 DPKG 관리 시스템 파일 설명서를 참조하세요.

RPM 패키지 관리자(RPM)

Amazon Linux 또는 Red Hat Enterprise Linux와 같이 RPM 패키지 관리자(yum/dnf)를 사용하는 컨테이너는 컨테이너 빌드 중에 RPM 데이터베이스가 제거되지 않았는지 확인해야 합니다. 자세한 내용은 RPM 웹 사이트의 RPM 관리 시스템 파일 설명서를 참조하세요.

Amazon Inspector 스캔을 사용하여 사용자 지정 CI/CD 파이프라인 통합 생성

CI/CD 솔루션에 Amazon Inspector CI/CD 플러그인을 사용할 수 있는 경우 Amazon Inspector CI/CD 플러그인을 사용하는 것이 좋습니다. CI/CD 솔루션에 Amazon Inspector CI/CD 플러그인을 사용할 수 없는 경우, Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 함께 사용하여 사용자 지정 CI/CD 통합을 생성할 수 있습니다. 다음 단계에서는 Amazon Inspector 스캔과 사용자 지정 CI/CD 파이프라인 통합을 생성하는 방법에 대해 설명합니다.

> **Tip**
>
> 단일 명령으로 SBOM을 생성하고 스캔하려는 경우 Amazon Inspector SBOM 생성기(Sbomgen)를 사용하여 3단계와 4단계를 건너뛸 수 있습니다.

1단계. 구성 AWS 계정

Amazon Inspector 스캔 API에 대한 액세스를 AWS 계정 제공하는를 구성합니다. 자세한 내용은 Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정 단원을 참조하십시오.

2단계. Sbomgen 바이너리 설치

Sbomgen 바이너리를 설치하고 구성합니다. 자세한 정보는 Sbomgen 설치를 참조하세요.

3단계. Sbomgen 사용하기

Sbomgen을 사용하여 스캔하려는 컨테이너 이미지에 대한 SBOM 파일을 생성합니다.

다음 예를 사용할 수 있습니다. *image:id*는 스캔할 이미지의 이름으로 바꿉니다. *sbom_path.json*은 SBOM 출력을 저장할 위치로 바꿉니다.

예제

```
./inspector-sbomgen container --image image:id -o sbom_path.json
```

4단계. Amazon Inspector 스캔 API 직접 호출

`inspector-scan` API를 직접적으로 호출하여 생성된 SBOM을 스캔하고 취약성 보고서를 제공합니다.

다음 예를 사용할 수 있습니다. *sbom_path.json*은 유효한 CycloneDX 호환 SBOM 파일의 위치로 바꿉니다. *ENDPOINT*를 AWS 리전 현재 인증된의 API 엔드포인트로 바꿉니다. *REGION*은 해당 리전으로 바꿉니다.

예제

```
aws inspector-scan scan-sbom --sbom file://sbom_path.json --endpoint
ENDPOINT-URL --region REGION
```

AWS 리전 및 엔드포인트의 전체 목록은 리전 및 엔드포인트를 참조하세요.

(선택 사항) 5단계. 단일 명령으로 SBOM 생성 및 스캔

> **Note**
>
> 3단계와 4단계를 건너뛴 경우에만 이 단계를 완료하세요.

`--scan-bom` 플래그를 사용하여 단일 명령으로 SBOM을 생성하고 스캔합니다.

다음 예를 사용할 수 있습니다. *image:id*는 스캔할 이미지의 이름으로 바꿉니다. *profile*은 해당 프로파일로 바꿉니다. *REGION*은 해당 리전으로 바꿉니다. */tmp/scan.json*은 tmp 디렉터리에 있는 scan.json 파일의 위치로 바꿉니다.

예제

```
./inspector-sbomgen container --image image:id --scan-sbom --aws-profile
profile --aws-region REGION -o /tmp/scan.json
```

AWS 리전 및 엔드포인트의 전체 목록은 리전 및 엔드포인트를 참조하세요.

API 출력 형식

Amazon Inspector 스캔 API는 CycloneDX 1.5 형식 또는 Amazon Inspector 조사 결과 JSON 으로 취약성 보고서를 출력할 수 있습니다. `--output-format` 플래그를 사용하여 기본값을 변경할 수 있습니다.

CycloneDX 1.5 형식 출력의 예


```
{
  "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"
          }
        ]
      }
    ]
  }
}
```

Inspector 형식 출력 예

```
                        {
  "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"
          }
        ]
      }
    ]
  }
}
```


Amazon Inspector Jenkins 플러그인 사용

Jenkins 플러그인은 Amazon Inspector SBOM 생성기 바이너리와 Amazon Inspector 스캔 API를 사용하여 빌드 종료 시 상세한 보고서를 생성하므로 배포 전에 위험을 조사하고 해결할 수 있습니다. Amazon Inspector Jenkins 플러그인을 사용하면 Jenkins 파이프라인에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 또한 탐지된 취약성의 수와 심각도에 따라 파이프라인 실행에 성공하거나 실패하도록 Amazon Inspector 취약성 스캔을 구성할 수 있습니다. Jenkins 플러그인의 최신 버전은 Jenkins 마켓플레이스(https://plugins.jenkins.io/amazon-inspector-image-scanner/)에서 확인할 수 있습니다. 다음 단계에서는 Amazon Inspector Jenkins 플러그인을 설정하는 방법에 대해 설명합니다.

> **Important**
>
> 다음 단계를 완료하기 전에 플러그인을 실행하려면 Jenkins를 버전 2.387.3 이상으로 업그레이드해야 합니다.

1단계. 설정 AWS 계정

Amazon Inspector 스캔 API에 대한 액세스를 허용하는 IAM 역할 AWS 계정 로를 구성합니다. 지침은 Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정 섹션을 참조하세요.

2단계. Amazon Inspector Jenkins 플러그인을 설치합니다.

다음 절차에서는 Jenkins 대시보드에서 Amazon Inspector Jenkins 플러그인을 설치하는 방법에 대해 설명합니다.

1. Jenkins 대시보드에서 Jenkins 관리를 선택한 다음 플러그인 관리를 선택합니다.
2. 사용 가능을 선택합니다.
3. 사용 가능 탭에서 Amazon Inspector 스캔을 검색한 다음 플러그인을 설치합니다.

(선택 사항) 3단계. Jenkins에 docker 자격 증명 추가

> **Note**
>
> Docker 이미지가 프라이빗 리포지토리에 있는 경우에만 docker 자격 증명을 추가합니다. 그렇지 않은 경우 이 단계를 건너뜁니다.

다음 절차에서는 Jenkins 대시보드에서 Jenkins에 docker 자격 증명을 추가하는 방법에 대해 설명합니다.

1. Jenkins 대시보드에서 Manage Jenkins, Credentials, System을 차례로 선택합니다.
2. Global credentials를 선택한 다음 Add credentials를 선택합니다.
3. Kind에서 Username with password를 선택합니다.
4. Scope에서 Global (Jenkins, nodes, items, all child items, etc)을 선택합니다.
5. 세부 정보를 입력한 다음 OK를 선택합니다.

(선택 사항) 4단계. AWS 자격 증명 추가

> **Note**
>
> IAM 사용자를 기반으로 인증하려는 경우에만 AWS 자격 증명을 추가합니다. 그렇지 않은 경우 이 단계를 건너뜁니다.

다음 절차에서는 Jenkins 대시보드에서 AWS 자격 증명을 추가하는 방법을 설명합니다.

1. Jenkins 대시보드에서 Manage Jenkins, Credentials, System을 차례로 선택합니다.
2. Global credentials를 선택한 다음 Add credentials를 선택합니다.
3. Kind에서 AWS Credentials를 선택합니다.
4. Access Key ID 및 Secret Access Key를 포함한 세부 정보를 입력한 다음 OK를 선택합니다.

5단계. Jenkins 스크립트에 CSS 지원 추가

다음 절차에서는 Jenkins 스크립트에 CSS 지원을 추가하는 방법에 대해 설명합니다.

1. Jenkins를 다시 시작합니다.
2. 대시보드에서 Manage Jenkins, Nodes, Built-In Node, Script Console을 차례로 선택합니다.
3. 텍스트 상자에 `System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")` 줄을 추가한 다음 Run을 선택합니다.

6단계. 빌드에 Amazon Inspector 스캔 추가

프로젝트에 빌드 단계를 추가하거나 Jenkins 선언적 파이프라인을 사용하여 Amazon Inspector 스캔을 빌드에 추가할 수 있습니다.

프로젝트에 빌드 단계를 추가하여 빌드에 Amazon Inspector 스캔 추가

1. 구성 페이지에서 Build Steps까지 아래로 스크롤하여 Add build step을 선택합니다. 그런 다음 Amazon Inspector Scan을 선택합니다.

2. 두 가지 inspector-sbomgen 설치 방법(Automatic 또는 Manual) 중에서 선택합니다. 자동 옵션을 사용하면 플러그인이 최신 버전을 다운로드할 수 있습니다. 또한 항상 최신 특성, 보안 업데이트 및 버그 수정이 있는지 확인합니다.

   a. (옵션 1) 최신 버전의 Inspector-sbomgen을 다운로드하려면 Automatic을 선택합니다. 이 옵션은 현재 사용 중인 운영 체제 및 CPU 아키텍처를 자동으로 감지합니다.

   b. (옵션 2) 스캔을 위해 Amazon Inspector SBOM 생성기 바이너리를 설정하려면 Manual을 선택합니다. 이 방법을 선택하는 경우 이전에 다운로드한 Inspector-sbomgen 버전의 전체 경로를 제공해야 합니다.

   자세한 내용은 Amazon Inspector SBOM 생성기에서 Amazon Inspector SBOM 생성기(Sbomgen) 설치를 참조하세요.

3. 다음을 완료하여 Amazon Inspector 스캔 빌드 단계 구성을 완료합니다.

   a. 이미지 ID를 입력합니다. 이미지는 로컬, 원격 또는 아카이브 위치에 있을 수 있습니다. 이미지 이름은 Docker 이름 지정 규칙을 따라야 합니다. 내보낸 이미지를 분석하는 경우 예상 tar 파일의 경로를 제공하세요. 다음 이미지를 예시 이미지 ID 경로로 참조하세요.

      i. 로컬 또는 원격 컨테이너의 경우: `NAME[:TAG|@DIGEST]`

      ii. tar 파일의 경우: `/path/to/image.tar`

   b. 스캔 요청을 보낼 AWS 리전을 선택합니다.

   c. (선택 사항) 보고서 아티팩트 이름에 빌드 프로세스 중에 생성된 아티팩트의 사용자 지정 이름을 입력합니다. 이를 통해 이를 고유하게 식별하고 관리할 수 있습니다.

   d. (선택 사항) 파일 건너뛰기에서 스캔에서 제외할 디렉터리를 하나 이상 지정합니다. 크기로 인해 스캔할 필요가 없는 디렉터리의 경우 이 옵션을 고려하세요.

   e. (선택 사항) Docker 보안 인증 정보의 경우 Docker 사용자 이름을 선택합니다. 컨테이너 이미지가 프라이빗 리포지토리에 있는 경우에만 이 작업을 수행하세요.

   f. (선택 사항) 다음과 같이 지원되는 AWS 인증 방법을 제공할 수 있습니다.

      i. (선택 사항) IAM role에 역할 ARN(arn:aws:iam::*AccountNumber*:role/*RoleName*)을 제공합니다.

      ii. (선택 사항) AWS 자격 증명에서 IAM 사용자를 기반으로 인증할 자격 AWS 증명을 지정합니다.

      iii. (선택 사항) AWS profile name에 프로파일 이름을 사용하여 인증할 프로파일의 이름을 입력합니다.

g. (선택 사항) 취약성 임계값 활성화를 선택합니다. 이 옵션을 사용하면 스캔한 취약성이 값을 초과하는 경우 빌드가 실패하는지 여부를 확인할 수 있습니다. 모든 값이 0과 같으면 스캔되는 취약성 수에 관계없이 빌드가 성공합니다. EPSS 점수의 경우 값은 0~1일 수 있습니다. 스캔한 취약성이 값을 초과하면 빌드가 실패하고 EPSS 점수가 값보다 높은 모든 CVE가 콘솔에 표시됩니다.

4. 저장을 선택합니다.

Jenkins 선언적 파이프라인을 사용하여 빌드에 Amazon Inspector 스캔 추가

Jenkins 선언적 파이프라인을 사용하여 자동 또는 수동으로 빌드에 Amazon Inspector 스캔을 추가할 수 있습니다.

SBOMGen 선언적 파이프라인을 자동으로 다운로드하려면

- 빌드에 Amazon Inspector 스캔을 추가하려면 다음 예제 구문을 사용합니다. *IMAGE_PATH*는 이미지 경로(예: *alpine:latest*)로, *IAM_ROLE*은 1단계에서 구성한 IAM 역할의 ARN으로, *ID*는 Docker 자격 증명 ID(프라이빗 리포지토리를 사용하는 경우)로 바꿉니다. 선택적으로 취약성 임계값을 활성화하고 각 심각도에 대한 값을 지정할 수 있습니다.

```
pipeline {
    agent any
    stages {
        stage('amazon-inspector-image-scanner') {
            steps {
                script {
                step([
                $class:
 'com.amazon.inspector.jenkins.amazoninspectorbuildstep.AmazonInspectorBuilder',
                archivePath: 'IMAGE_PATH', // Path to your container image or tar file
                awsRegion: 'REGION', // AWS region for scan requests
                iamRole: 'IAM ROLE', // IAM role ARN for authentication
                credentialId: 'Id', // Docker credentials (empty if public repo)
                awsCredentialId: 'AWS ID', // AWS credential ID for authentication
                awsProfileName: 'Profile Name', // AWS profile name to use
                sbomgenSkipFiles: '*.log,node_modules,/tmp/*', // Files/directories to
 exclude from scanning

                // Vulnerability threshold settings (updated parameter names)
```

```
                isSeverityThresholdEnabled: false, // Enable/disable build failure on
 vulnerability count
                countCritical: 0, // Max critical vulnerabilities before build fails
                countHigh: 0, // Max high vulnerabilities before build fails
                countMedium: 5, // Max medium vulnerabilities before build fails
                countLow: 10, // Max low vulnerabilities before build fails

                // EPSS (Exploit Prediction Scoring System) settings
                isEpssThresholdEnabled: false, // Enable/disable EPSS-based failure
 threshold
                epssThreshold: 0.7, // EPSS score threshold (0.0 to 1.0)

                // NEW FEATURE: CVE Suppression - ignore specific false positives
                isSuppressedCveEnabled: false, // Enable CVE suppression feature
                suppressedCveList: '', // Comma-separated list of CVEs to ignore in
 thresholds

                // NEW FEATURE: Auto-Fail CVEs - always fail on critical security
 issues
                isAutoFailCveEnabled: false, // Enable auto-fail CVE feature
                autoFailCveList: '' // Comma-separated list of CVEs that always fail
 build
                ])
            }
        }
    }
}
```

SBOMGen 선언적 파이프라인을 수동으로 다운로드하려면

- 빌드에 Amazon Inspector 스캔을 추가하려면 다음 예제 구문을 사용합니다. *SBOMGEN_PATH*는 3단계에서 설치한 Amazon Inspector SBOM 생성기의 경로로, *IMAGE_PATH*는 이미지 경로(예: *alpine:latest*)로, *IAM_ROLE*은 1단계에서 구성한 IAM 역할의 ARN으로, *ID*는 Docker 자격 증명 ID(프라이빗 리포지토리를 사용하는 경우)로 바꿉니다. 선택적으로 취약성 임계값을 활성화하고 각 심각도에 대한 값을 지정할 수 있습니다.

> **Note**
>
> Sbomgen을 Jenkins 디렉터리에 배치하고 플러그인의 Jenkins 디렉터리 경로를 제공합니다(예: */opt/folder/arm64/inspector-sbomgen*).

```
pipeline {
    agent any
    stages {
        stage('amazon-inspector-image-scanner') {
            steps {
                script {
                step([
                $class:
 'com.amazon.inspector.jenkins.amazoninspectorbuildstep.AmazonInspectorBuilder',
                archivePath: 'IMAGE_PATH', // Path to your container image or tar file
                awsRegion: 'REGION', // AWS region for scan requests
                iamRole: 'IAM ROLE', // IAM role ARN for authentication
                credentialId: 'Id', // Docker credentials (empty if public repo)
                awsCredentialId: 'AWS ID', // AWS credential ID for authentication
                awsProfileName: 'Profile Name', // AWS profile name to use
                sbomgenSkipFiles: '*.log,node_modules,/tmp/*', // Files/directories to
 exclude from scanning

                // Vulnerability threshold settings (updated parameter names)
                isSeverityThresholdEnabled: false, // Enable/disable build failure on
 vulnerability count
                countCritical: 0, // Max critical vulnerabilities before build fails
                countHigh: 0, // Max high vulnerabilities before build fails
                countMedium: 5, // Max medium vulnerabilities before build fails
                countLow: 10, // Max low vulnerabilities before build fails

                // EPSS (Exploit Prediction Scoring System) settings
                isEpssThresholdEnabled: false, // Enable/disable EPSS-based failure
 threshold
                epssThreshold: 0.7, // EPSS score threshold (0.0 to 1.0)

                // NEW FEATURE: CVE Suppression - ignore specific false positives
                isSuppressedCveEnabled: false, // Enable CVE suppression feature
                suppressedCveList: '', // Comma-separated list of CVEs to ignore in
 thresholds

                // NEW FEATURE: Auto-Fail CVEs - always fail on critical security
 issues
                isAutoFailCveEnabled: false, // Enable auto-fail CVE feature
                autoFailCveList: '' // Comma-separated list of CVEs that always fail
 build
                ])
            }
```

```
        }
    }
}
```

플러그인에는 보안 취약성을 관리하기 위한 특성이 포함되어 있습니다.

억제된 CVE 목록

스캔은 때때로 실제 위협이 아닌 취약성을 탐지할 수 있습니다. 이러한 오탐이 빌드를 중지하지 않도록 하려면 표시되지 않는 목록에 추가할 수 있습니다.

```
isSuppressedCveEnabled: true,
suppressedCveList: 'CVE-2023-1234,CVE-2023-5678'
```

이렇게 하면 빌드가 실패하는지 확인할 때 특정 CVE는 무시됩니다. 오탐을 해결한 경우에만 억제 목록에 오탐을 추가해야 합니다. 이러한 취약성을 금지 목록에 추가한 후에도 CVE는 여전히 보안 보고서에 표시되지만 빌드 실패는 발생하지 않습니다.

자동 실패 CVE 목록

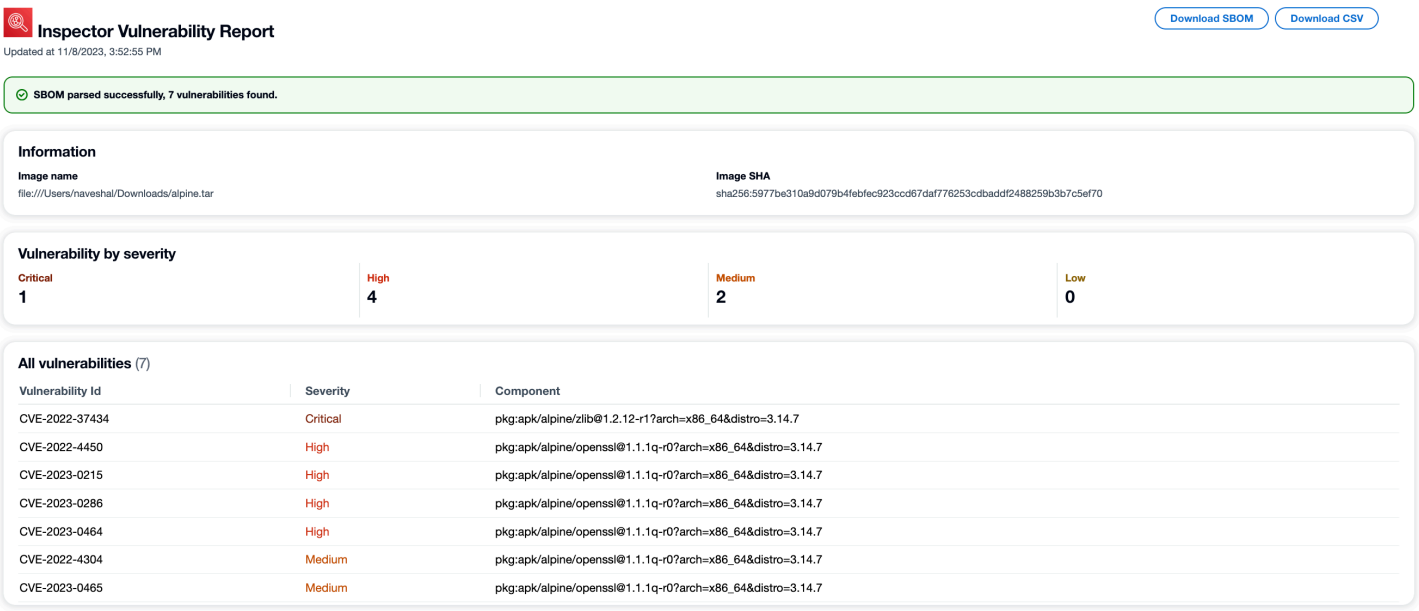
중요한 보안 취약성의 경우 항상 빌드가 실패하는 목록을 생성할 수 있습니다.

```
isAutoFailCveEnabled: true,
autoFailCveList: 'CVE-2024-9999'
```

이렇게 하면 활성화한 설정에 관계없이 항상 빌드가 실패합니다. 배포해서는 안 되는 우선 순위가 높은 보안 문제에 대해서만 이 목록을 생성해야 합니다. 이 목록은 보안을 극대화하기 위해 다른 모든 임계값 설정을 재정의합니다.

7단계. Amazon Inspector 취약성 보고서 확인

1. 프로젝트의 새 빌드를 완료합니다.
2. 빌드가 완료되면 결과에서 출력 형식을 선택합니다. HTML을 선택하면 보고서의 JSON SBOM 또는 CSV 버전을 다운로드할 수 있는 옵션이 제공됩니다. 다음은 HTML 보고서의 예입니다.



Note

플러그인은 이전 파라미터 이름을 지원하므로 이전 스크립트를 사용할 수 있습니다. 그러나 콘솔에서 이러한 파라미터를 최신 파라미터로 업데이트하라는 경고가 표시됩니다. 예를 들어 `isThresholdEnabled`를 사용하는 경우 파라미터를 `isSeverityThresholdEnabled`로 업데이트하라는 경고가 표시됩니다.

문제 해결

다음은 Jenkins용 Amazon Inspector 스캔 플러그인을 사용할 때 발생할 수 있는 몇 가지 일반적인 오류입니다.

자격 증명 로드 실패 또는 sts 예외 오류

오류:

`InstanceProfileCredentialsProvider(): Failed to load credentials or sts exception.`

해결 방법

AWS 계정에 `aws_secret_access_key` 대해 `aws_access_key_id` 및를 가져옵니다. `aws_access_key_id` 및 `aws_secret_access_key`를 `~/.aws/credentials`에 설정합니다.

tarball, 로컬 또는 원격 소스에서 이미지 로드 실패

오류:

```
2024/10/16 02:25:17 [ImageDownloadFailed]: failed to load image from
tarball, local, or remote sources.
```

> **Note**
>
> 이 오류는 Jenkins 플러그인이 컨테이너 이미지를 읽을 수 없거나, Docker 엔진에서 컨테이너 이미지를 찾을 수 없거나, 원격 컨테이너 레지스트리에서 컨테이너 이미지를 찾을 수 없는 경우에 발생할 수 있습니다.

해결 방법:

다음 사항을 확인합니다.

- Jenkins 플러그인 사용자에게 스캔하려는 이미지에 대한 읽기 권한이 있습니다.
- 스캔하려는 이미지가 Docker 엔진에 있습니다.
- 원격 이미지 URL이 정확합니다.
- 원격 레지스트리에 대해 인증되었습니다(해당하는 경우).

Inspector-sbomgen 경로 오류

오류:

```
Exception:com.amazon.inspector.jenkins.amazoninspectorbuildstep.exception.Sbomge
There was an issue running inspector-sbomgen, is /opt/inspector/inspector-
sbomgen the correct path?
```

해결 방법:

다음 절차에 따라 문제를 해결합니다.

1. Jenkins 디렉터리에 올바른 OS 아키텍처 Inspector-sbomgen을 배치합니다. 자세한 내용은 Amazon Inspector SBOM 생성기를 참조하세요.
2. `chmod +x inspector-sbomgen` 명령을 사용하여 바이너리에 실행 권한을 부여합니다.

3. 플러그인에 올바른 Jenkins 머신 경로를 제공합니다(예: `/opt/folder/arm64/inspector-sbomgen`).
4. 구성을 저장하고 Jenkins 작업을 실행합니다.

Amazon Inspector TeamCity 플러그인 사용

Amazon Inspector TeamCity 플러그인은 Amazon Inspector SBOM 생성기 바이너리와 Amazon Inspector 스캔 API를 사용하여 빌드 종료 시 상세한 보고서를 생성하므로 배포 전에 위험을 조사하고 해결할 수 있습니다. Amazon Inspector TeamCity 플러그인을 사용하면 TeamCity 파이프라인에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 또한 탐지된 취약성의 수와 심각도에 따라 파이프라인 실행에 성공하거나 실패하도록 Amazon Inspector 취약성 스캔을 구성할 수 있습니다. Amazon Inspector TeamCity 플러그인의 최신 버전은 TeamCity 마켓플레이스(https://plugins.jetbrains.com/plugin/23236-amazon-inspector-scanner)에서 확인할 수 있습니다. Amazon Inspector 스캔을 CI/CD 파이프라인에 통합하는 자세한 방법은 Amazon Inspector 스캔을 CI/CD 파이프라인에 통합을 참조하세요. Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어 목록은 지원되는 운영 체제 및 프로그래밍 언어를 참조하세요. 다음 단계에서는 Amazon Inspector TeamCity 플러그인을 설정하는 방법에 대해 설명합니다.

1. 를 설정합니다 AWS 계정.
   - Amazon Inspector 스캔 API에 대한 액세스를 허용하는 IAM 역할 AWS 계정 로를 구성합니다. 지침은 Amazon Inspector CI/CD 통합을 사용하도록 AWS 계정 설정 섹션을 참조하세요.
2. Amazon Inspector TeamCity 플러그인을 설치합니다.
   a. 대시보드에서 관리 > 플러그인으로 이동합니다.
   b. Amazon Inspector 스캔을 검색합니다.
   c. 플러그인을 설치합니다.
3. Amazon Inspector SBOM 생성기를 설치합니다.
   - Teamcity 서버 디렉터리에 Amazon Inspector SBOM 생성기 바이너리를 설치합니다. 지침은 Sbomgen 설치하기 섹션을 참조하세요.
4. Amazon Inspector 스캔 빌드 단계를 프로젝트에 추가합니다.
   a. 구성 페이지에서 Build Steps까지 아래로 스크롤하여 Add build step을 선택한 다음 Amazon Inspector Scan을 선택합니다.
   b. 다음 세부 정보를 입력하여 Amazon Inspector 스캔 빌드 단계를 구성합니다.

- Step name을 추가합니다.
- 두 가지 Amazon Inspector SBOM 생성기 설치 방법(Automatic 또는 Manual) 중에서 선택합니다.
  - Automatic은 시스템 및 CPU 아키텍처에 따라 최신 버전의 Amazon Inspector SBOM 생성기를 다운로드합니다.
  - Manual을 사용하려면 이전에 다운로드한 Amazon Inspector SBOM 생성기 버전의 전체 경로를 제공해야 합니다.

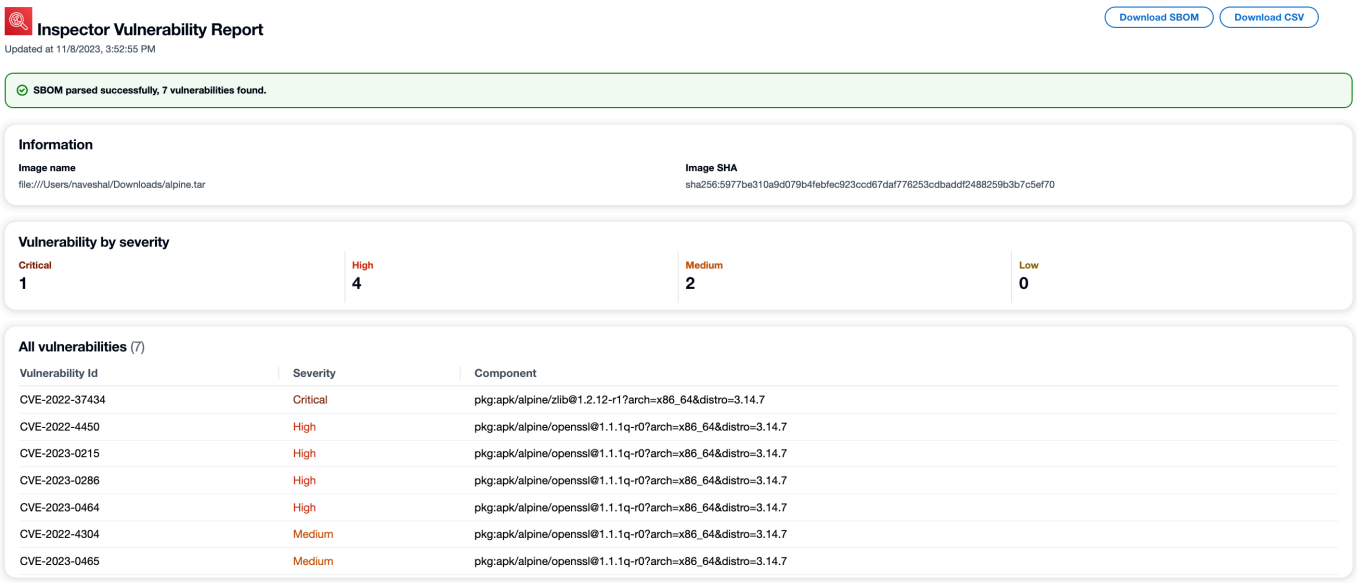
  자세한 내용은 Amazon Inspector SBOM 생성기에서 Amazon Inspector SBOM 생성기 (Sbomgen) 설치를 참조하세요.
- 이미지 ID를 입력합니다. 이미지는 로컬, 원격 또는 아카이브 위치에 있을 수 있습니다. 이미지 이름은 Docker 이름 지정 규칙을 따라야 합니다. 내보낸 이미지를 분석하는 경우 예상 tar 파일의 경로를 제공하세요. 다음 이미지를 예시 이미지 ID 경로로 참조하세요.
  - 로컬 또는 원격 컨테이너의 경우: `NAME[:TAG|@DIGEST]`
  - tar 파일의 경우: `/path/to/image.tar`
- IAM 역할의 경우 1단계에서 구성한 역할의 ARN을 입력합니다.
- 스캔 요청을 보낼 AWS 리전을 선택합니다.
- (선택 사항) Docker 인증의 경우 Docker 사용자 이름과 Docker 암호를 입력합니다. 컨테이너 이미지가 프라이빗 리포지토리에 있는 경우에만 이 작업을 수행하세요.
- (선택 사항) AWS 인증에 AWS 액세스 키 ID와 AWS 보안 키를 입력합니다. 자격 AWS 증명을 기반으로 인증하려는 경우에만이 작업을 수행합니다.
- (선택 사항) 심각도별 취약성 임계값을 지정합니다. 스캔 중에 지정한 수를 초과하면 이미지 빌드가 실패합니다. 값이 모두 `0`이면 발견되는 취약성 수와 상관없이 빌드가 성공합니다.

c. 저장을 선택합니다.

5. Amazon Inspector 취약성 보고서를 확인합니다.

   a. 프로젝트의 새 빌드를 완료합니다.

   b. 빌드가 완료되면 결과에서 출력 형식을 선택합니다. HTML을 선택하면 보고서의 JSON SBOM 또는 CSV 버전을 다운로드할 수 있습니다. 다음은 HTML 보고서의 예입니다.



GitHub 작업과 함께 Amazon Inspector 사용

Amazon Inspector를 GitHub actions과 함께 사용하여 GitHub 워크플로에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 이 플러그인은 Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 사용하여 빌드 종료 시 상세한 보고서를 생성하므로 배포 전에 위험을 조사하고 해결할 수 있습니다. 또한 탐지된 취약성의 수와 심각도에 따라 워크로드를 성공하거나 실패하도록 Amazon Inspector 취약성 스캔을 구성할 수 있습니다. Amazon Inspector 작업의 최신 버전은 GitHub 웹 사이트에서 확인할 수 있습니다. Amazon Inspector 스캔을 CI/CD 파이프라인에 통합하는 자세한 방법은 Amazon Inspector 스캔을 CI/CD 파이프라인에 통합을 참조하세요. Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어 목록은 지원되는 운영 체제 및 프로그래밍 언어를 참조하세요.

GitLab 구성 요소와 함께 Amazon Inspector 사용

Amazon Inspector를 GitLab CI/CD 구성 요소와 함께 사용하여 GitLab 프로젝트에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 이 플러그인은 Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 사용하여 빌드 종료 시 상세한 보고서를 생성하므로 배포 전에 위험을 조사하고 해결할 수 있습니다. 또한 탐지된 취약성의 수와 심각도에 따라 워크로드를 성공하거나 실패하도록 Amazon Inspector 취약성 스캔을 구성할 수 있습니다. Amazon Inspector 구성 요소의 최신 버전은 GitLab 웹 사이트에서 확인할 수 있습니다. Amazon Inspector 스캔을 CI/CD 파이프라인에 통합하는 자세한 방법은 Amazon Inspector 스캔을 CI/CD 파이프라인에 통합을 참조하세요. Amazon Inspector

에서 지원하는 운영 체제 및 프로그래밍 언어 목록은 지원되는 운영 체제 및 프로그래밍 언어를 참조하세요.

Amazon Inspector와 함께 CodeCatalyst 작업 사용

Amazon Inspector를 Amazon CodeCatalyst와 함께 사용하여 CodeCatalyst 워크플로에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 이 플러그인은 Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 사용하여 빌드 종료 시 상세한 보고서를 생성하므로 배포 전에 위험을 조사하고 해결할 수 있습니다. 또한 탐지된 취약성의 수와 심각도에 따라 워크로드를 성공하거나 실패하도록 Amazon Inspector 취약성 스캔을 구성할 수 있습니다. Amazon Inspector 스캔을 CI/CD 파이프라인에 통합하는 자세한 방법은 Amazon Inspector 스캔을 CI/CD 파이프라인에 통합을 참조하세요. Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어 목록은 지원되는 운영 체제 및 프로그래밍 언어를 참조하세요.

CodePipeline에서 Amazon Inspector 스캔 작업 사용

워크플로 AWS CodePipeline 에 취약성 스캔을 추가하여에서 Amazon Inspector를 사용할 수 있습니다. 이 통합을 통해 Amazon Inspector SBOM 생성기와 Amazon Inspector 스캔 API를 사용하여 빌드 종료 시 상세한 보고서를 생성합니다. 통합을 통해 배포 전에 위험을 조사하고 해결할 수 있습니다. `InspectorScan` 작업은 CodePipeline의 관리형 컴퓨팅 작업으로, 오픈 소스 코드의 보안 취약성을 자동으로 탐지하고 수정합니다. 이 작업은 GitHub 또는 Bitbucket Cloud와 같은 타사 리포지토리의 애플리케이션 소스 코드 또는 컨테이너 애플리케이션용 이미지와 함께 사용할 수 있습니다. 자세한 내용은 AWS CodePipeline 사용 설명서의 InspectorScan 간접 호출 작업 참조를 참조하세요.

AWS 환경의 Amazon Inspector 적용 범위 평가

Amazon Inspector 콘솔의 계정 관리 화면에서 AWS 환경의 Amazon Inspector 적용 범위를 평가할 수 있습니다.이 화면에는 계정 및 리소스에 대한 Amazon Inspector 스캔 상태에 대한 세부 정보와 통계가 표시됩니다.

> **Note**
>
> 조직의 위임된 관리자인 경우 조직의 모든 계정에 대한 세부 정보와 통계를 볼 수 있습니다.

다음 절차에서는 Amazon Inspector 환경의 적용 범위를 평가하는 방법에 대해 설명합니다.

AWS 환경의 Amazon Inspector 적용 범위를 평가하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 계정 관리를 선택합니다.
3. 적용 범위를 검토하려면 다음 탭 중 하나를 선택합니다.
   - 계정 수준의 적용 범위를 검토하려면 계정을 선택합니다.
   - Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 대한 적용 범위를 검토하려면 인스턴스를 선택합니다.
   - Amazon Elastic Container Registry(Amazon ECR) 리포지토리에 대한 적용 범위를 검토하려면 컨테이너 리포지토리를 선택합니다.
   - Amazon ECR 컨테이너 이미지에 대한 적용 범위를 검토하려면 컨테이너 이미지를 선택합니다.
   - Lambda 함수에 대한 적용 범위를 검토하려면 Lambda 함수를 선택합니다.

다음 주제에서는 이러한 각 탭이 제공하는 정보에 대해 설명합니다.

주제

- 계정 수준 적용 범위 평가
- Amazon EC2 인스턴스의 적용 범위 평가
- Amazon ECR 리포지토리의 적용 범위 평가
- Amazon ECR 컨테이너 이미지의 적용 범위 평가

- AWS Lambda 함수의 적용 범위 평가

계정 수준 적용 범위 평가

계정이 조직의 일부가 아니거나 조직의 위임된 Amazon Inspector 관리자 계정이 아닌 경우 계정 탭에는 사용자 계정에 대한 정보와 계정의 리소스 스캔 상태가 제공됩니다. 이 탭에서는 계정의 모든 리소스 또는 특정 유형의 리소스에 대해서만 스캔을 활성화하거나 비활성화할 수 있습니다. 자세한 내용은 Amazon Inspector의 자동 스캔 유형 단원을 참조하십시오.

계정이 조직의 위임된 Amazon Inspector 관리자 계정인 경우, 계정 탭에는 조직의 계정에 대한 자동 활성화 설정이 제공되며 조직의 모든 계정이 나열됩니다. 각 계정에 대해 목록에는 해당 계정에 대해 Amazon Inspector가 활성화되어 있는지 여부와 활성화된 경우 계정에 활성화된 리소스 스캔 유형이 표시됩니다. 위임된 관리자는 이 탭에서 조직의 자동 활성화 설정을 변경할 수 있습니다. 개별 멤버 계정에 대해 특정 유형의 리소스 검색을 활성화하거나 비활성화할 수도 있습니다. 자세한 내용은 멤버 계정에 대한 Amazon Inspector 스캔 활성화 단원을 참조하십시오.

Amazon EC2 인스턴스의 적용 범위 평가

인스턴스 탭에는 AWS 환경의 Amazon EC2 인스턴스가 표시됩니다. 목록은 다음 탭에서 그룹으로 구성되어 있습니다.

- 모두 - 사용자 환경에 있는 모든 인스턴스가 표시됩니다. 상태 열은 인스턴스의 현재 스캔 상태를 나타냅니다.
- 스캔 – Amazon Inspector가 사용자 환경에서 능동적으로 모니터링 및 스캔하고 있는 모든 인스턴스가 표시됩니다.
- 스캔하지 않음 – Amazon Inspector가 사용자 환경에서 모니터링 및 스캔하지 않는 모든 인스턴스가 표시됩니다. 이유 열은 Amazon Inspector가 인스턴스를 모니터링 및 스캔하지 않는 이유를 나타냅니다.

  여러 가지 이유로 EC2 인스턴스가 스캔하지 않음 탭에 표시될 수 있습니다. Amazon Inspector는 AWS Systems Manager (SSM) 및 SSM 에이전트를 사용하여 EC2 인스턴스의 취약성을 자동으로 모니터링하고 스캔합니다. 인스턴스에 실행 중인 SSM 에이전트가 없거나, Systems Manager를 지원하는 AWS Identity and Access Management (IAM) 역할이 없거나, 지원되는 운영 체제 또는 아키텍처를 실행하지 않는 경우 Amazon Inspector는 인스턴스를 모니터링하고 스캔할 수 없습니다. 자세한 내용은 Amazon EC2 인스턴스 스캔 단원을 참조하십시오.

각 탭에서 계정 열은 인스턴스를 소유 AWS 계정 하는를 지정합니다.

EC2 인스턴스 태그 - 이 열에는 인스턴스와 연결된 태그가 표시되며, 이를 통해 해당 인스턴스가 태그로 인해 스캔에서 제외되었는지 확인할 수 있습니다.

운영 체제 - 이 열에는 운영 체제 유형(`WINDOWS`, `MAC`, `LINUX` 또는 `UNKNOWN`)이 표시됩니다.

다음을 사용하여 모니터링 - 이 열에는 Amazon Inspector가 이 인스턴스에 대해 에이전트 기반 스캔 방법을 사용하는지 아니면 에이전트 없는 스캔 방법을 사용하는지 여부가 표시됩니다.

마지막 스캔 - 이 열에는 Amazon Inspector가 해당 리소스의 취약성을 마지막으로 검사한 시간이 표시됩니다. Amazon Inspector에서 스캔을 수행하는 빈도는 인스턴스를 스캔하는 데 사용하는 스캔 방법에 따라 다릅니다.

EC2 인스턴스에 대한 추가 세부 정보를 검토하려면 EC2 인스턴스 열의 링크를 선택하세요. 그러면 Amazon Inspector에서 인스턴스에 대한 세부 정보와 해당 인스턴스에 대한 현재 조사 결과를 표시합니다. 결과에 대한 세부 정보를 검토하려면 제목 열의 링크를 선택하세요. 이러한 세부 정보에 대한 자세한 내용은 Amazon Inspector 조사 결과에 대한 세부 정보 보기 섹션을 참조하세요.

Amazon EC2 인스턴스의 스캔 상태 값

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 경우 가능한 상태 값은 다음과 같습니다.

- 능동 모니터링 – Amazon Inspector가 인스턴스를 지속적으로 모니터링하고 스캔합니다.
- 에이전트 없는 인스턴스 스토리지 한도 초과 - 인스턴스에 연결된 모든 볼륨의 합산 크기가 1200GB를 초과하거나 인스턴스에 연결된 볼륨이 8개 이상인 경우 Amazon Inspector에서 이 상태를 사용합니다.
- 에이전트 없는 인스턴스 수집 시간 한도 초과 - 인스턴스에 대해 에이전트 없는 스캔을 실행하는 동안 Amazon Inspector의 제한 시간이 초과되었습니다.
- EC2 인스턴스가 중지됨 – 인스턴스가 중지 상태이기 때문에 Amazon Inspector에서 인스턴스 검색을 일시 중지했습니다. 기존 조사 결과는 인스턴스가 종료될 때까지 보존됩니다. 인스턴스가 다시 시작되면 Amazon Inspector가 자동으로 인스턴스 스캔을 재개합니다.
- 내부 오류 – Amazon Inspector에서 인스턴스 스캔을 시도할 때 내부 오류가 발생했습니다. Amazon Inspector에서 자동으로 오류를 해결하고 가능한 한 빨리 스캔을 재개합니다.
- 인벤토리 없음 – Amazon Inspector에서 인스턴스를 스캔할 소프트웨어 애플리케이션 인벤토리를 찾지 못했습니다. 인스턴스의 Amazon Inspector 연결이 삭제되었거나 실행에 실패했을 수 있습니다.

  이 문제를 해결하려면 AWS Systems Manager 를 사용하여 `InspectorInventoryCollection-do-not-delete` 연결이 존재하고 연결 상태가 성공적인지 확인합니다. 또한 AWS Systems

Manager Fleet Manager를 사용하여 인스턴스에 대한 소프트웨어 애플리케이션 인벤토리를 확인하세요.

- 비활성화 보류 중 – Amazon Inspector가 인스턴스 스캔을 중단했습니다. 인스턴스는 비활성화되어 정리 작업이 완료될 때까지 대기 중입니다.
- 첫 번째 스캔 보류 중 – Amazon Inspector에서 첫 번째 스캔을 위해 인스턴스를 대기열에 추가했습니다.
- 리소스 종료됨 – 인스턴스가 종료되었습니다. Amazon Inspector가 현재 인스턴스에 대한 기존 조사 결과 및 적용 범위 데이터를 정리하고 있습니다.
- 기한 경과 인벤토리 – Amazon Inspector에서 인스턴스에 대해 지난 7일 이내에 캡처한 업데이트된 소프트웨어 애플리케이션 인벤토리를 수집하지 못했습니다.

  이 문제를 해결하려면 AWS Systems Manager 를 사용하여 필요한 Amazon Inspector 연결이 존재하고 인스턴스에 대해 실행 중인지 확인합니다. 또한 AWS Systems Manager Fleet Manager를 사용하여 인스턴스에 대한 소프트웨어 애플리케이션 인벤토리를 확인하세요.
- 비관리형 EC2 인스턴스 – Amazon Inspector에서 인스턴스를 모니터링하거나 스캔하지 않습니다. AWS Systems Manager에서 인스턴스를 관리하지 않습니다.

  이 문제를 해결하려면 AWS Systems Manager Automation에서 AWSSupport-TroubleshootManagedInstance runbook 제공하는를 사용할 수 있습니다. 인스턴스를 관리 AWS Systems Manager 하도록를 구성한 후 Amazon Inspector는 인스턴스를 자동으로 지속적으로 모니터링하고 스캔하기 시작합니다.
- 지원되지 않는 OS – Amazon Inspector에서 인스턴스를 모니터링하거나 스캔하지 않습니다. 인스턴스가 Amazon Inspector에서 지원하지 않는 운영 체제 또는 아키텍처를 사용합니다. Amazon Inspector에서 지원하는 운영 체제 목록은 Amazon EC2 인스턴스 상태 값 섹션을 참조하세요.
- 능동적으로 모니터링, 부분적 오류 발생) - 이 상태는 EC2 스캔이 활성화되어 있지만 Linux 기반 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사와 관련된 오류가 있음을 의미합니다. 가능한 심층 검사 오류는 다음과 같습니다.
  - 심층 검사 패키지 수집 한도 초과 – 인스턴스가 Amazon Inspector 심층 검사에 대한 패키지 한도인 5000개를 초과했습니다. 이 인스턴스의 심층 검사를 재개하려면 계정과 관련된 사용자 지정 경로를 조정해 볼 수 있습니다.
  - 심층 검사 일일 SSM 인벤토리 한도 초과 – 이 인스턴스에 대해 매일 인스턴스당 수집되는 인벤토리 데이터의 SSM 할당량에 이미 도달했기 때문에 SSM 에이전트에서 Amazon Inspector로 인벤토리를 보낼 수 없습니다. 자세한 내용은 Amazon EC2 Systems Manager 엔드포인트 및 할당량을 참조하세요.

- 심층 검사 수집 시간 제한 초과 – 패키지 수집 시간이 최대 임곗값인 15분을 초과하여 Amazon Inspector에서 패키지 인벤토리를 추출하지 못했습니다.
- 심층 검사에 인벤토리가 없음 – Amazon Inspector SSM 플러그인에서 이 인스턴스의 패키지 인벤토리를 아직 수집하지 못했습니다. 이는 일반적으로 보류 중인 스캔의 결과이지만, 6시간 후에도 이 상태가 지속되면 Amazon EC2 Systems Manager를 사용하여 해당 인스턴스에 필요한 Amazon Inspector 연결이 존재하고 실행 중인지 확인하세요.

EC2 인스턴스의 스캔 설정 구성에 대한 자세한 내용은 Amazon EC2 인스턴스 스캔 섹션을 참조하세요.

Amazon ECR 리포지토리의 적용 범위 평가

리포지토리 탭에는 AWS 환경의 Amazon ECR 리포지토리가 표시됩니다. 목록은 다음 탭에서 그룹으로 구성되어 있습니다.

- 모두 - 사용자 환경에 있는 모든 리포지토리가 표시됩니다. 상태 열은 리포지토리의 현재 스캔 상태를 나타냅니다.
- 활성화됨 – Amazon Inspector가 사용자 환경에서 모니터링 및 스캔하도록 구성된 모든 리포지토리가 표시됩니다. 상태 열은 리포지토리의 현재 스캔 상태를 나타냅니다.
- 활성화 안됨 – Amazon Inspector가 사용자 환경에서 모니터링 및 스캔하지 않는 모든 리포지토리가 표시됩니다. 이유 열은 Amazon Inspector가 리포지토리를 모니터링 및 스캔하지 않는 이유를 나타냅니다.

각 탭에서 계정 열은 리포지토리를 소유 AWS 계정 한를 지정합니다.

리포지토리에 대한 추가 세부 정보를 검토하려면 리포지토리 이름을 선택하세요. 그러면 Amazon Inspector가 리포지토리에 있는 컨테이너 이미지 목록과 각 이미지에 대한 세부 정보를 표시합니다. 세부 정보에는 이미지 태그, 이미지 다이제스트 및 스캔 상태가 포함됩니다. 또한 이미지의 중요 조사 결과 수와 같은 주요 결과 통계도 포함됩니다. 조사 결과 통계에 대한 지원 데이터를 드릴다운하여 검토하려면 이미지의 이미지 태그를 선택하세요.

> **Note**
>
> 연속 스캔이 없는 Amazon ECR 이미지는 적용 범위 위젯에 포함되지 않습니다.

Amazon ECR 리포지토리의 스캔 상태 값

Amazon Elastic Container Registry(Amazon ECR) 리포지토리의 경우 가능한 상태 값은 다음과 같습니다.

- 활성화됨(연속) – 활성화됨(연속) - 리포지토리의 경우 Amazon Inspector에서 이 리포지토리에 있는 이미지를 지속적으로 모니터링합니다. 리포지토리의 고급 스캔 설정이 연속 스캔으로 설정되어 있습니다. Amazon Inspector는 새 이미지가 푸시되면 처음에 이미지를 스캔하고 해당 이미지와 관련된 새로운 CVE가 게시되면 이미지를 다시 스캔합니다. Amazon Inspector는 사용자가 구성한 Amazon ECR 재스캔 기간 동안 이 리포지토리의 이미지를 계속 모니터링합니다.
- 활성화됨(푸시 시) – 새 이미지가 푸시될 때 Amazon Inspector에서 리포지토리의 개별 컨테이너 이미지를 자동으로 스캔합니다. 리포지토리에 대해 고급 스캔이 활성화되어 있고 푸시할 때 스캔으로 설정되어 있습니다.
- 액세스 거부됨 – Amazon Inspector에서 리포지토리 또는 리포지토리에 있는 컨테이너 이미지에 액세스할 수 없습니다.

  이 문제를 해결하려면 리포지토리에 대한 AWS Identity and Access Management (IAM) 정책이 Amazon Inspector가 리포지토리에 액세스할 수 있도록 허용하는지 확인합니다.
- 비활성화됨(수동) – Amazon Inspector에서 리포지토리에 있는 컨테이너 이미지를 모니터링하거나 스캔하지 않습니다. 리포지토리의 Amazon ECR 스캔 설정이 기본(수동 스캔)으로 설정되어 있습니다.

  Amazon Inspector로 리포지토리 이미지 스캔을 시작하려면 리포지토리의 스캔 설정을 고급 스캔으로 변경한 다음 이미지를 지속적으로 스캔할지 아니면 새 이미지가 푸시될 때만 스캔할지 여부를 선택하세요.
- 활성화됨(푸시 시) – 새 이미지가 푸시될 때 Amazon Inspector에서 리포지토리의 개별 컨테이너 이미지를 자동으로 스캔합니다. 리포지토리의 고급 스캔 설정이 푸시할 때 스캔으로 설정되어 있습니다.
- 내부 오류 – Amazon Inspector에서 리포지토리를 스캔하려고 할 때 내부 오류가 발생했습니다. Amazon Inspector에서 자동으로 오류를 해결하고 가능한 한 빨리 스캔을 재개합니다.

리포지토리에 대한 스캔 설정 구성에 대한 자세한 내용은 Amazon ECR 컨테이너 이미지 스캔 단원을 참조하세요.

Amazon ECR 컨테이너 이미지의 적용 범위 평가

이미지 탭에는 AWS 환경의 Amazon ECR 컨테이너 이미지가 표시됩니다. 목록은 다음 탭에서 그룹으로 구성되어 있습니다.

- 모두 – 사용자 환경에 있는 모든 컨테이너 이미지가 표시됩니다. 상태 열은 이미지의 현재 스캔 상태를 나타냅니다.
- 스캔 – Amazon Inspector가 사용자 환경에서 모니터링 및 스캔하도록 구성된 모든 컨테이너 이미지가 표시됩니다. 상태 열은 이미지의 현재 스캔 상태를 나타냅니다.
- 스캔하지 않음 – Amazon Inspector가 사용자 환경에서 모니터링 및 스캔하지 않는 모든 컨테이너 이미지가 표시됩니다. 이유 열은 Amazon Inspector가 이미지를 모니터링 및 스캔하지 않는 이유를 나타냅니다.

  여러 가지 이유로 컨테이너 이미지가 활성화되지 않음 탭에 표시될 수 있습니다. 이미지가 Amazon Inspector 스캔이 활성화되지 않은 리포지토리에 저장되어 있거나 Amazon ECR 필터링 규칙으로 인해 해당 리포지토리가 스캔되지 않을 수 있습니다. 또는 ECR 재스캔 기간으로 설정한 기간 내에 이미지를 푸시하거나 가져오지 않은 경우입니다. 자세한 내용은 Amazon ECR 재스캔 기간 구성을 참조하세요.

각 탭의 리포지토리 이름 열은 컨테이너 이미지를 저장하는 리포지토리의 이름을 지정합니다. 계정 열은 리포지토리를 소유 AWS 계정 하는를 지정합니다. 마지막 스캔 열에는 Amazon Inspector가 해당 리소스의 취약성을 마지막으로 검사한 시간이 표시됩니다. 여기에는 조사 결과 메타데이터에 대한 업데이트가 있을 때, 리소스의 애플리케이션 인벤토리에 대한 업데이트가 있을 때, 새로운 CVE에 대한 대응으로 재스캔이 진행될 때 수행되는 검사가 포함될 수 있습니다. 자세한 내용은 Amazon ECR 스캔의 스캔 동작 단원을 참조하십시오.

컨테이너 이미지에 대한 추가 세부 정보를 검토하려면 ECR 컨테이너 이미지 열의 링크를 선택하세요. 그러면 Amazon Inspector에서 이미지에 대한 세부 정보와 이미지에 대한 현재 조사 결과를 표시합니다. 조사 결과에 대한 세부 정보를 검토하려면 제목 열의 링크를 선택하세요. 이러한 세부 정보에 대한 자세한 내용은 Amazon Inspector 조사 결과에 대한 세부 정보 보기 섹션을 참조하세요.

Amazon ECR 컨테이너 이미지의 스캔 상태 값

Amazon Elastic Container Registry 컨테이너 이미지의 경우 가능한 상태 값은 다음과 같습니다.

- 능동 모니터링(연속) - Amazon Inspector에서 지속적으로 이미지를 모니터링하고 있으며, 관련 CVE가 새로 게시될 때마다 이미지에 대한 새로운 스캔이 수행됩니다. 이미지를 푸시하거나 가져올 때마

다 이미지의 Amazon ECR 재스캔 기간이 새로 고쳐집니다. 이미지를 저장하는 리포지토리에 대해 고급 스캔이 활성화되고 리포지토리의 고급 스캔 설정이 연속 스캔으로 설정되어 있습니다.

- 활성화됨(푸시 시) – 새 이미지가 푸시될 때마다 Amazon Inspector에서 이미지를 자동으로 스캔합니다. 이미지를 저장하는 리포지토리에 대해 고급 스캔이 활성화되고 리포지토리의 고급 스캔 설정이 푸시할 때 스캔으로 설정되어 있습니다.
- 내부 오류 – Amazon Inspector에서 컨테이너 이미지를 스캔하려고 할 때 내부 오류가 발생했습니다. Amazon Inspector에서 자동으로 오류를 해결하고 가능한 한 빨리 스캔을 재개합니다.
- 첫 번째 스캔 보류 중 – Amazon Inspector에서 첫 번째 스캔을 위해 이미지를 대기열에 추가했습니다.
- 스캔 자격 만료됨(연속) - Amazon Inspector에서 이미지 스캔을 일시 중단했습니다. 리포지토리 이미지의 자동 재스캔 기간 내에 이미지가 업데이트되지 않았습니다. 이미지를 푸시하거나 가져와서 스캔을 재개할 수 있습니다.
- 스캔 자격 만료됨(푸시 시) - Amazon Inspector에서 이미지 스캔을 일시 중단했습니다. 리포지토리 이미지의 자동 재스캔 기간 내에 이미지가 업데이트되지 않았습니다. 이미지를 푸시하여 스캔을 재개할 수 있습니다.
- 스캔 빈도 수동(수동) – Amazon Inspector에서 Amazon ECR 컨테이너 이미지를 스캔하지 않습니다. 이미지를 저장하는 리포지토리의 Amazon ECR 스캔 설정이 기본(수동 스캔)으로 설정되어 있습니다. Amazon Inspector로 리포지토리 이미지 자동 스캔을 시작하려면 리포지토리 설정을 고급 스캔으로 변경한 다음 이미지를 지속적으로 스캔할지 아니면 새 이미지가 푸시될 때만 스캔할지 여부를 선택하세요.
- 지원되지 않는 OS – Amazon Inspector에서 인스턴스를 모니터링하거나 스캔하지 않습니다. 이미지가 Amazon Inspector에서 지원하지 않는 운영 체제를 기반으로 하거나 Amazon Inspector에서 지원하지 않는 미디어 유형을 사용합니다.

  Amazon Inspector에서 지원하는 운영 체제 목록은 지원되는 운영 체제: Amazon Inspector를 사용한 Amazon ECR 스캔 섹션을 참조하세요. Amazon Inspector에서 지원하는 미디어 유형 목록은 지원되는 미디어 유형을 참조하세요.

리포지토리 및 이미지의 스캔 설정 구성에 대한 자세한 내용은 Amazon ECR 컨테이너 이미지 스캔 섹션을 참조하세요.

AWS Lambda 함수의 적용 범위 평가

Lambda 탭에는 AWS 환경의 Lambda 함수가 표시됩니다. 이 페이지에는 두 개의 테이블이 있는데, 하나는 Lambda 표준 스캔에 대한 함수 적용 범위 세부 정보를 보여주고 다른 하나는 Lambda 코드 스캔에 대한 함수 적용 범위 세부 정보를 보여줍니다. 다음 탭을 기준으로 함수를 그룹화할 수 있습니다.

- 모두 - 사용자 환경에 있는 모든 Lambda 함수가 표시됩니다. 상태 열은 Lambda 함수의 현재 스캔 상태를 나타냅니다.
- 스캔 – Amazon Inspector가 스캔하도록 구성된 Lambda 함수가 표시됩니다. 상태 열은 각 Lambda 함수의 현재 스캔 상태를 나타냅니다.
- 스캔하지 않음 - Amazon Inspector가 스캔하도록 구성되지 않은 Lambda 함수가 표시됩니다. 이유 열은 Amazon Inspector가 함수를 모니터링 및 스캔하지 않는 이유를 나타냅니다.

  여러 가지 이유로 Lambda 함수가 스캔하지 않음 탭에 표시될 수 있습니다. Lambda 함수가 Amazon Inspector에 추가되지 않은 계정에 속해 있거나 필터링 규칙으로 인해 이 함수가 스캔되지 않을 수 있습니다. 자세한 내용은 Lambda 함수 스캔 단원을 참조하십시오.

각 탭의 함수 이름 열은 Lambda 함수의 이름을 지정합니다. 계정 열은 함수를 소유 AWS 계정 하는를 지정합니다. 런타임은 함수의 런타임을 지정합니다. 상태 열은 각 Lambda 함수의 현재 스캔 상태를 나타냅니다. 리소스 태그에는 함수에 적용된 태그가 표시됩니다. 마지막 스캔 열에는 Amazon Inspector가 해당 리소스의 취약성을 마지막으로 검사한 시간이 표시됩니다. 여기에는 조사 결과 메타데이터에 대한 업데이트가 있을 때, 리소스의 애플리케이션 인벤토리에 대한 업데이트가 있을 때, 새로운 CVE에 대한 대응으로 재스캔이 진행될 때 수행되는 검사가 포함될 수 있습니다. 자세한 내용은 Lambda 함수 스캔의 스캔 동작 단원을 참조하십시오.

AWS Lambda 함수의 상태 값 스캔

Lambda 함수의 경우 가능한 상태 값은 다음과 같습니다.

- 능동 모니터링 – Amazon Inspector에서 Lambda 함수를 지속적으로 모니터링하고 스캔합니다. 연속 스캔에는 새 함수를 리포지토리로 푸시할 때의 최초 스캔과 함수가 업데이트되거나 새로운 일반 취약성 및 노출(CVE)이 발표될 때 자동으로 함수를 재스캔하는 것이 포함됩니다.
- 태그를 기준으로 제외됨 – 해당 함수는 태그 기준 스캔에서 제외되었으므로 Amazon Inspector에서 스캔하지 않습니다.
- 스캔 자격 만료됨 – 해당 함수가 마지막으로 간접적으로 호출되거나 업데이트된 지 90일 이상 지났기 때문에 Amazon Inspector에서 이 함수를 모니터링하지 않습니다.

- 내부 오류 – Amazon Inspector에서 함수 스캔을 시도할 때 내부 오류가 발생했습니다. Amazon Inspector에서 자동으로 오류를 해결하고 가능한 한 빨리 스캔을 재개합니다.
- 첫 번째 스캔 보류 중 – Amazon Inspector에서 첫 번째 스캔을 위해 함수를 대기열에 추가했습니다.
- 지원되지 않음 – Lambda 함수에 지원되지 않는 런타임이 있습니다.

를 사용하여 Amazon Inspector에서 여러 계정 관리 AWS Organizations

Amazon Inspector를 사용하여 조직의 여러 계정을 관리할 수 있습니다. Amazon Inspector는 다중 계정 관리를 위한 두 가지 접근 방식을 지원합니다.

- AWS Organizations 정책의 위임된 관리자 - 여러 리전의 조직 계정에서 Amazon Inspector를 자동으로 활성화하여 위임된 관리자에게 중앙 집중식 거버넌스를 제공합니다. 조직 정책은 활성화된 스캔 유형을 적용하고 정책 관리형이 아닌 위임된 관리자 및 멤버 계정 활성화보다 우선합니다.
- 비 AWS Organizations 정책에 대한 위임된 관리자 - 조직 정책을 사용하지 않고 조직의 Amazon Inspector를 관리하도록 지정된 계정입니다. 위임된 관리자는 멤버 계정에 대해 Amazon Inspector를 활성화하고 스캔 설정을 구성할 수 있습니다.

이러한 접근 방식을 함께 사용할 수 있습니다. 조직 정책이 적용되면 리소스 유형 활성화(사용 가능한 스캔 유형)를 제어하는 반면, 위임된 관리자는 스캔 모드 및 심층 검사 경로와 같은 스캔 구성 설정을 제어합니다. 다음 주제에서는 이러한 관리 접근 방식, 위임된 관리자를 지정하는 방법, 멤버 계정을 관리하는 방법을 설명합니다.

주제

- Amazon Inspector의 위임된 관리자 계정 및 멤버 계정 이해
- Amazon Inspector 위임된 관리자 계정 지정

Amazon Inspector의 위임된 관리자 계정 및 멤버 계정 이해

Amazon Inspector를 다중 계정 환경에서 사용하는 경우 위임된 관리자 계정으로 특정 메타데이터에 액세스할 수 있습니다. 메타데이터에는 Amazon EC2, Amazon ECR 및 Lambda에 대한 표준 스캔과 Lambda 코드 스캔이 포함됩니다. 또한 멤버 계정에 대한 보안 조사 결과도 포함됩니다. 이 단원에서는 위임된 관리자 계정이 수행할 수 있는 작업과 멤버 계정이 수행할 수 있는 작업에 대한 정보를 제공합니다.

조직 정책 거버넌스 모델

AWS Organizations 정책을 사용하여 Amazon Inspector를 활성화하면 허용되는 작업을 결정하는 거버넌스 모델이 적용됩니다.

정책 관리형 리소스

조직 정책에 의해 명시적으로 활성화 또는 비활성화된 리소스는 위임된 관리자 또는 멤버 계정에서 수정할 수 없습니다. 정책 관리형 스캔 유형을 활성화 또는 비활성화하기 위한 API 요청은 리소스가 조직 정책에 의해 관리됨을 나타내는 명확한 오류와 함께 실패합니다.

Non-policy-managed 리소스

조직 정책에 지정되지 않은 리소스는 위임된 관리자 및 멤버 계정에서 Amazon Inspector 콘솔 또는 API를 사용하여 정상적으로 관리할 수 있습니다.

스캔 구성 관리

위임된 관리자는 리소스 유형이 정책 관리형인지 여부에 관계없이 EC2 스캔 모드, 심층 검사 경로 및 ECR 재스캔 기간과 같은 스캔 설정을 항상 구성할 수 있습니다. 조직 정책은 스캔이 활성화되었는지 여부만 제어하고 작동 방식은 제어하지 않습니다.

Amazon Inspector 조직 정책 생성 및 관리에 대한 자세한 내용은 Amazon Inspector 정책 AWS Organizations 설명서를 참조하세요.

위임 관리자 작업

일반적으로 위임된 관리자가 자신의 계정에 설정을 적용하면 해당 설정이 조직의 다른 모든 계정에 적용됩니다. 또한 위임 관리자는 자신의 계정 및 연결된 멤버의 정보를 보고 검색할 수 있습니다. Amazon Inspector 위임 관리자 계정은 다음 작업을 수행할 수 있습니다.

- AWS Organizations 관리 계정만 위임된 관리자를 지정하고 제거할 수 있습니다.
- 위임된 관리자를 지정할 때는 관리하려는 멤버 계정과 동일한 조직에 있어야 합니다.
- Amazon Inspector 활성화 및 비활성화를 포함하여 관련 계정의 Amazon Inspector 상태를 확인하고 관리합니다.
- 조직 내 모든 멤버 계정의 스캔 유형을 활성화하거나 비활성화합니다.
- 조직 전체에서 집계된 조사 결과 데이터와 조직 내 모든 멤버 계정에 대한 결과 세부 정보를 확인합니다.
- 조직 내 모든 계정의 조사 결과에 적용될 억제 규칙을 생성하고 관리합니다.
- 조직의 모든 멤버에 대해 Amazon ECR 고급 스캔을 활성화합니다.
- 전체 조직의 리소스 적용 범위를 확인합니다.
- 조직 내 모든 멤버 계정에 대해 ECR 컨테이너 이미지 자동 재스캔 기간을 정의합니다. 위임 관리자의 스캔 기간 설정이 이전에 멤버 계정이 설정한 모든 설정보다 우선합니다. 조직의 모든 계정은 위

임된 관리자의 Amazon ECR 자동 재스캔 기간을 공유합니다. 계정마다 다른 재스캔 기간을 설정할 수 없습니다.

- Amazon EC2용 Amazon Inspector 심층 검사에 대해 조직의 모든 계정에서 사용할 5개의 사용자 지정 경로를 지정합니다. 이는 위임 관리자가 개별 계정에 대해 설정할 수 있는 5개의 사용자 지정 경로에 추가하여 지정하는 것입니다. 심층 검사 사용자 지정 경로 구성에 대한 자세한 내용은 Amazon Inspector 심층 검사를 위한 사용자 지정 경로 단원을 참조하세요.
- 멤버 계정에 대한 Amazon Inspector 심층 검사를 활성화 및 비활성화합니다.
- 조직 내 모든 멤버 계정의 SBOM을 내보냅니다.
- 조직의 모든 멤버 계정에 Amazon EC2 스캔 모드를 설정합니다. 자세한 내용은 스캔 모드 관리 단원을 참조하십시오.
- 멤버 계정에서 생성한 스캔 구성을 제외하고 조직의 모든 계정에 대해 CIS 스캔 구성을 생성하고 관리합니다.

> **Note**
>
> 멤버 계정이 조직에서 탈퇴하면 위임된 관리자는 더 이상 해당 계정에서 예약한 스캔 구성을 볼 수 없게 됩니다.

- 조직의 모든 계정에 대한 CIS 스캔 결과를 확인합니다.
- 조직 정책을 사용하는 경우 정책 관리형 리소스에 대한 스캔 설정을 구성하지만 정책 관리형 스캔 유형 자체를 활성화하거나 비활성화할 수는 없습니다.

멤버 계정 작업

멤버 계정은 Amazon Inspector에서 자신의 계정에 대한 정보를 보고 검색할 수 있지만, 계정 설정은 위임된 관리자가 관리합니다. 조직 내 멤버 계정은 Amazon Inspector에서 다음 작업을 수행할 수 있습니다.

- 본인 계정에 대해 Amazon Inspector를 활성화합니다.
- 본인 계정에 대한 리소스 적용 범위를 확인합니다.
- 본인 계정에 대한 조사 결과 세부 정보를 확인합니다.
- 본인 계정에 대한 ECR 컨테이너 이미지 자동 재스캔 기간 설정을 확인합니다.
- Amazon Inspector의 EC2 심층 검사를 위해 개별 계정에 사용할 5개의 사용자 지정 경로를 지정합니다. 이 경로는 위임된 관리자가 조직에 대해 지정한 사용자 지정 경로에 추가하여 스캔됩니다. 심층

검사 경로 구성에 대한 자세한 내용은 Amazon Inspector 심층 검사를 위한 사용자 지정 경로 단원을 참조하세요.

- Amazon Inspector 심층 검사를 위해 위임된 관리자가 설정한 사용자 지정 경로를 확인합니다.
- 자신의 계정과 연결된 리소스의 SBOM을 내보냅니다.
- 자신의 계정의 스캔 모드를 확인합니다.
- 자신의 계정에 대한 CIS 스캔 구성을 생성하고 관리합니다.
- 위임된 관리자가 예약한 리소스를 포함하여 계정의 리소스에 대한 CIS 스캔 결과를 확인합니다.
- 조직 정책에서 관리하지 않는 스캔 유형을 활성화합니다. 정책 관리형 스캔 유형은 멤버 계정에서 활성화하거나 비활성화할 수 없습니다.

> **Note**
>
> Amazon Inspector는 활성화 후 위임 관리자 계정을 통해서만 비활성화할 수 있습니다.

Amazon Inspector 위임된 관리자 계정 지정

위임된 관리자는 조직에 대한 서비스를 관리하는 계정입니다. 이 주제에서는 Amazon Inspector 위임된 관리자를 지정하는 방법에 대해 설명합니다.

고려 사항

위임된 관리자를 지정하기 전에 다음 사항에 유의하세요.

위임된 관리자는 최대 10,000명의 멤버를 관리할 수 있습니다.

멤버 계정이 10,000개를 초과하는 경우 Amazon CloudWatch Personal Health Dashboard를 통해 그리고 위임된 관리자 계정으로 이메일을 통해 알림을 받게 됩니다.

> **Note**
>
> 10,000개 이상의 계정(최대 50,000개)이 있는 조직에 대한 AWS Organizations 정책을 통해 Amazon Inspector가 활성화된 경우이 정책은 모든 계정에 적용됩니다. 그러나 Amazon Inspector 조직에는 10,000개의 계정만 연결됩니다. 즉, 위임된 관리자는 Amazon Inspector 콘솔에서 이러한 10,000개의 계정에 대한 조사 결과 및 계정 상태만 볼 수 있습니다.

위임 관리자는 리전별로 결정됩니다.

Amazon Inspector는 리전 서비스입니다. Amazon Inspector를 사용하려는 모든 AWS 리전 에서 절차의 단계를 반복해야 합니다.

조직의 위임 관리자는 한 명입니다.

한에서 계정을 위임된 관리자로 지정하는 경우 AWS 리전해당 계정은 다른 모든에서 위임된 관리자여야 합니다 AWS 리전.

위임 관리자를 변경해도 멤버 계정의 Amazon Inspector는 비활성화되지 않습니다.

위임된 관리자를 제거해도 멤버 계정은 독립 실행형 계정이 되며 스캔 설정은 영향을 받지 않습니다.

AWS 조직에 모든 기능이 활성화되어 있어야 합니다.

의 기본 설정입니다 AWS Organizations. 활성화되어 있지 않은 경우 조직 내 모든 특성 활성화를 참조하세요.

조직 정책은 위임된 관리자 설정보다 우선합니다.

조직에서 AWS Organizations 정책을 사용하여 Amazon Inspector를 활성화하는 경우 정책 설정에 따라 활성화된 스캔 유형이 결정됩니다. 일관된 거버넌스를 보장하기 위해 조직 정책을 생성하기 전에 위임된 관리자를 지정하는 것이 좋습니다. 자세한 내용은 조직 정책 거버넌스 모델 단원을 참조하십시오.

위임 관리자를 지정하는 데 필요한 권한

Amazon Inspector를 활성화하고 Amazon Inspector 위임 관리자를 지정할 수 있는 권한이 있어야 합니다. 이러한 권한을 부여하려면 IAM 정책의 끝에 다음 문을 추가합니다. 자세한 내용은 IAM 정책 관리를 참조하세요.

```
{
    "Sid": "PermissionsForInspectorAdmin",
    "Effect": "Allow",
    "Action": [
        "inspector2:EnableDelegatedAdminAccount",
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:ListDelegatedAdministrators",
```

```
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization"
    ],
    "Resource": "*"
}
```

AWS 조직의 위임된 관리자 지정

다음 절차는 조직의 위임 관리자를 지정하는 방법을 보여줍니다. 절차를 완료하기 전에 위임된 관리자가 관리할 멤버 계정과 동일한 조직에 있는지 확인합니다.

> **Note**
>
> 이 절차를 완료하려면 AWS Organizations 관리 계정을 사용해야 합니다. AWS Organizations 관리 계정만 위임된 관리자를 지정할 수 있습니다. 위임된 관리자를 지정하려면 권한이 필요할 수 있습니다. 자세한 내용은 위임 관리자를 지정하는 데 필요한 권한 단원을 참조하십시오.

Amazon Inspector를 처음 활성화하면 Amazon Inspector가 해당 계정에 대한 서비스 연결 역할 `AWSServiceRoleForAmazonInspector`를 생성합니다. Amazon Inspector에서 서비스 연결 역할을 사용하는 방법에 대한 자세한 내용은 Amazon Inspector에 서비스 연결 역할 사용 섹션을 참조하세요.

Console

Amazon Inspector 위임 관리자를 지정하려면

1. AWS Organizations 관리 계정에 로그인한 다음 https://console.aws.amazon.com/inspector/v2/home Amazon Inspector 콘솔을 엽니다.
2. AWS 리전 선택기를 사용하여 위임된 관리자를 지정할 AWS 리전 을 지정합니다.
3. 탐색 창에서 일반 설정을 선택합니다.
4. 위임된 관리자에서 위임된 관리자로 AWS 계정 지정할의 12자리 ID를 입력합니다.
5. 위임을 선택한 다음 위임을 다시 선택합니다.

위임된 관리자를 지정하면 기본적으로 계정에 대해 모든 스캔 유형이 활성화됩니다. AWS Organizations 관리 계정에 대해 Amazon Inspector를 활성화하려면 다음 절차를 완료하세요.

AWS Organizations 관리 계정에 대해 Amazon Inspector를 활성화하려면

1. 위임된 관리자 계정에 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 탐색 창에서 계정 관리를 선택합니다.
3. 계정에서 AWS Organizations 관리 계정을 선택한 다음 활성화를 선택합니다.
4. AWS Organizations 관리 계정에 대해 활성화하려는 스캔 유형을 선택한 다음 제출을 선택합니다.

API

API를 사용하여 위임된 관리자 지정

- AWS 계정 Organizations 관리 계정의 보안 인증 정보를 사용하여 EnableDelegatedAdminAccount API 작업을 실행합니다. AWS Command Line Interface를 사용하여 다음 CLI 명령을 실행하여이 작업을 수행할 수도 있습니다`aws inspector2 enable-delegated-admin-account --delegated-admin-account-id 11111111111`.

> **Note**
>
> Amazon Inspector 위임된 관리자로 지정할 계정의 계정 ID를 지정해야 합니다.

멤버 계정에 대한 Amazon Inspector 스캔 활성화

여러 방법을 통해 조직의 멤버 계정에 대해 Amazon Inspector를 활성화할 수 있습니다. 선택하는 방법은 거버넌스 요구 사항 및 조직 구조에 따라 다릅니다.

AWS Organizations 정책(중앙 집중식 거버넌스에 권장)

AWS Organizations 정책을 사용하여 중앙 집중식 제어를 통해 조직 전체에서 Amazon Inspector를 자동으로 활성화합니다. 이 접근 방식은 일관된 스캔 적용 범위를 보장하고 새 계정에 자동으로 적용됩니다. 자세한 지침은 Amazon Inspector 정책 생성 AWS Organizations 설명서를 참조하세요.

위임된 관리자 활성화

위임된 관리자는 Amazon Inspector 콘솔 또는 API를 통해 특정 멤버 계정 또는 모든 멤버 계정에 대해 Amazon Inspector를 수동으로 활성화할 수 있습니다. 이 접근 방식은 조직 정책을 사용하지 않을 때 유연성을 제공합니다.

멤버 계정 자체 활성화

멤버 계정은 조직 정책에 의해 제한되지 않는 경우 자신의 계정에 대해 Amazon Inspector를 활성화할 수 있습니다. 활성화되면 계정이 위임된 관리자와 연결됩니다.

멤버 계정 스캔 활성화

다음 절차에서는 위임된 관리자 및 멤버 계정 방법을 사용하여 멤버 계정에 대한 스캔을 활성화하는 방법을 설명합니다. Amazon Inspector 스캔 유형에 대한 자세한 내용은 Amazon Inspector의 자동 스캔 유형 단원을 참조하세요.

모든 멤버 계정에 대한 스캔을 자동으로 활성화하려면

1. 위임된 관리자 계정 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 리전 선택기를 사용하여 모든 멤버 계정에 대해 스캔을 활성화할 AWS 리전 를 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다. 계정 탭에는 AWS Organizations 관리 계정과 연결된 모든 멤버 계정이 표시됩니다.
4. 조직 아래에서 계정 번호 옆의 상자를 선택합니다. 그런 다음 활성화를 선택하여 멤버 계정에 적용할 스캔 옵션을 선택합니다. 다음 스캔 유형을 선택할 수 있습니다.

   - Amazon EC2 스캔
   - Amazon ECR 스캔
   - Lambda 표준 스캔
   - Lambda 코드 스캔

   - 원하는 스캔 유형을 선택한 후 저장을 선택합니다.

> **Note**
>
> 여러 페이지의 계정이 있는 경우 각 페이지에서 이 단계를 반복해야 합니다. 톱니바퀴 아이콘을 선택하여 각 페이지에 표시되는 계정 수를 변경할 수 있습니다.

5. 새 멤버 계정에 대해 Inspector 자동 활성화 설정을 켜고 조직에 추가된 새 멤버 계정에 적용할 스캔 옵션을 선택합니다. 다음 스캔 유형을 선택할 수 있습니다.

   - Amazon EC2 스캔
   - Amazon ECR 스캔
   - Lambda 표준 스캔
   - Lambda 코드 스캔

   - 원하는 스캔 유형을 선택한 후 활성화를 선택합니다.

   > **Note**
   >
   > 새 멤버 계정에 대해 자동으로 검사기 활성화 설정을 사용하면 조직의 이후 모든 멤버에 대해 Amazon Inspector가 활성화됩니다.
   > 멤버 계정 수가 5,000개를 초과할 경우 이 설정은 자동으로 꺼집니다. 총 멤버 계정 수가 5,000개 미만으로 줄어들면 이 설정은 자동으로 다시 활성화됩니다.

6. (권장) 멤버 계정에 대한 스캔을 활성화 AWS 리전 하려는 각에서 이러한 각 단계를 반복합니다.

특정 멤버 계정에 대한 스캔을 활성화하려면

1. 위임된 관리자 계정 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 리전 선택기를 사용하여 모든 멤버 계정에 대해 스캔을 활성화할 AWS 리전 를 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다. 계정 탭에는 AWS Organizations 관리 계정과 연결된 모든 멤버 계정이 표시됩니다.
4. 조직 아래에서 스캔을 활성화하려는 각 멤버 계정 번호 옆의 상자를 선택합니다. 그런 다음 활성화를 선택하여 멤버 계정에 적용할 스캔 옵션을 선택합니다. 다음 스캔 유형을 선택할 수 있습니다.

- Amazon EC2 스캔
- Amazon ECR 스캔
- Lambda 표준 스캔
- Lambda 코드 스캔

- 원하는 스캔 유형을 선택한 후 저장을 선택합니다.

> **Note**
>
> 여러 페이지의 계정이 있는 경우 각 페이지에서 이 단계를 반복해야 합니다. 톱니바퀴 아이콘을 선택하여 각 페이지에 표시되는 계정 수를 변경할 수 있습니다.

5. (권장) 특정 멤버에 대한 스캔을 활성화 AWS 리전 하려는 각에서 이러한 각 단계를 반복합니다.

멤버 계정으로 스캔을 활성화하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 리전 선택기를 사용하여 모든 멤버 계정에 대해 스캔을 활성화할 AWS 리전 를 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다. 계정 탭에는 AWS Organizations 관리 계정과 연결된 모든 멤버 계정이 표시됩니다.
4. 조직 아래에서 계정 번호 옆의 상자를 선택합니다. 그런 다음 활성화를 선택하여 적용할 스캔 옵션을 선택합니다. 다음 스캔 유형을 선택할 수 있습니다.

   - Amazon EC2 스캔
   - Amazon ECR 스캔
   - Lambda 표준 스캔
   - Lambda 코드 스캔

   - 원하는 스캔 유형을 선택한 후 저장을 선택합니다.
5. (권장) 멤버 계정에 대한 스캔을 활성화하려는 리전마다 위 단계를 반복합니다.

> **Note**
>
> AWS Organizations 관리 계정에 Amazon Inspector에 대한 위임된 관리자 계정이 있는 경우 멤버 계정으로 계정을 활성화하여 스캔 세부 정보를 볼 수 있습니다.

> **중요**
>
> 조직 정책이 계정에 대한 Amazon Inspector 활성화를 관리하는 경우 위임된 관리자 및 멤버 계정은 Amazon Inspector 활성화/비활성화 APIs를 사용하여 정책 관리형 스캔 유형을 수정할 수 없습니다. API 요청이 실패하고 리소스가 조직 정책에 의해 관리됨을 나타내는 오류가 발생합니다. 정책에서 관리하지 않는 추가 스캔 유형은 계속 활성화할 수 있습니다.

Amazon Inspector에서 멤버 계정 연결 해제

위임된 관리자는 계정에서 멤버 계정의 연결을 해제해야 할 수 있습니다. 멤버 계정 연결을 해제해도 Amazon Inspector는 계정에서 계속 활성화되며 해당 계정은 독립 실행형 계정이 됩니다. 또한 더 이상 계정에 대한 Amazon Inspector 관리 권한이 없습니다. 하지만 이전에 연결이 해제된 멤버 계정을 언제든지 내 계정과 연결할 수 있습니다. 이 단원에서는 위임된 관리자로서 멤버 계정 연결을 해제하는 방법에 대해 설명합니다.

> **Note**
>
> 정책 관리형 계정의 연결을 해제하려면 스캔 유형에 대해 해당 계정에 연결된 Amazon Inspector 조직 정책이 없어야 합니다.

Console

콘솔을 사용하여 멤버 계정의 연결을 해제하는 방법

1. 위임된 관리자 계정 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 리전 선택기를 사용하여 멤버 계정의 연결을 해제할 AWS 리전 를 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다.
4. 조직 아래에서 연결을 해제할 각 계정 번호 옆의 상자를 선택합니다.

5. 작업을 선택한 다음 계정 연결 해제를 선택합니다.

API

API를 사용하여 멤버 계정의 연결을 해제하는 방법

DisassociateMember API 작업을 실행합니다. 요청에 연결을 해제하려는 계정 ID를 입력합니다.

Amazon Inspector에서 위임된 관리자 제거

Amazon Inspector 위임된 관리자 계정을 제거해야 할 수 있습니다. AWS Organizations 관리 계정에서 이 작업을 수행할 수 있습니다. Amazon Inspector 위임된 관리자 계정을 제거해도 해당 계정과 모든 멤버 계정에서 Amazon Inspector가 계속 활성화됩니다. 위임된 관리자 계정과 모든 멤버 계정은 독립 실행형 계정이 되어 원래의 스캔 설정을 유지합니다.

> **Note**
>
> AWS Organizations 정책이 Amazon Inspector 활성화를 관리하는 경우 위임된 관리자를 제거해도 정책 적용에는 영향을 주지 않습니다. 멤버 계정 조사 결과는 새로운 위임된 관리자가 지정될 때까지 중앙 위임된 관리자 콘솔에 더 이상 표시되지 않지만, 계정은 조직 정책 설정에 따라 활성화된 상태로 유지됩니다.

이 단원에서는 위임된 관리자 계정을 제거하는 방법에 대해 설명합니다.

Amazon Inspector 위임된 관리자 제거

다음 절차에서는 Amazon Inspector 위임된 관리자를 제거하는 방법과 위임된 관리자 계정에서 멤버 계정을 연결하는 방법에 대해 설명합니다.

Amazon Inspector 위임된 관리자를 할당하는 자세한 방법은 Amazon Inspector 위임된 관리자 계정 지정을 참조하세요.

> **Note**
>
> Amazon Inspector 위임된 관리자를 할당한 후 Amazon Inspector 위임된 관리자는 멤버 계정을 수동으로 연결해야 합니다.

위임 관리자를 제거하려면

1. AWS Organizations 관리 계정을 AWS Management Console 사용하여에 로그인합니다.
2. Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
3. 리전 선택기를 사용하여 위임된 관리자를 제거할 AWS 리전 을 선택합니다.
4. 탐색 창에서 일반 설정을 선택합니다.
5. 위임된 관리자 섹션에서 제거를 선택한 다음 작업을 확인합니다.

멤버를 새 위임 관리자와 연결하려면

1. 위임된 관리자 계정 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 리전 선택기를 사용하여 멤버를 연결할 AWS 리전 을 선택합니다.
3. 탐색 창에서 계정 관리를 선택합니다.
4. 조직 아래에서 계정 번호 옆의 상자를 선택합니다.
5. 작업을 선택하고 멤버 추가를 선택합니다.

Amazon Inspector 리소스 태그 지정

태그는 AWS 리소스에 추가할 수 있는 레이블입니다. 태그를 사용하면 특정 기준에 따라 AWS 리소스를 분류할 수 있습니다. 태그는 키-값 쌍으로 이루어져 있습니다. 태그 키는 일반 레이블입니다. 태그 값은 태그 키에 대한 설명입니다. Amazon Inspector를 사용하면 억제 규칙 및 CIS 스캔 구성에 태그를 지정할 수 있습니다. 각 Amazon Inspector 리소스에 최대 50개의 태그를 추가할 수 있습니다.

태그 지정 기본 사항

태그는 키-값 쌍으로 이루어져 있습니다. 태그 키는 일반 레이블입니다. 태그 값은 태그 키에 대한 설명입니다. 이 주제에서는 Amazon Inspector 리소스에 태그를 지정하는 기본 사항을 설명합니다. Amazon Inspector 리소스에 태그를 지정할 때는 다음 사항을 고려하세요.

- 억제 규칙 및 CIS 스캔 구성에 태그를 지정할 수 있습니다.
- 각 Amazon Inspector 리소스에 최대 50개의 태그를 추가할 수 있습니다.
- 각 태그 키는 고유해야 합니다.
- 태그 키는 하나의 태그 값만 가질 수 있습니다.
- 태그 키와 태그 값은 최대 128개의 UTF-8 문자를 포함할 수 있습니다. 문자는 문자, 숫자, 공백 또는 `_ . : / = + - @` 기호일 수 있습니다.
- 어떤 태그에서도 `aws` 접두사를 사용하거나 이 접두사를 사용하여 태그를 수정할 수 없습니다. `aws` 접두사가 있는 태그는 AWS에서 사용하도록 예약되어 있습니다.
- Amazon Inspector 리소스에 할당된 태그는 AWS 계정과 해당 태그를 생성한 AWS 리전 에서만 사용할 수 있습니다.
- 리소스를 삭제하면 리소스에 연결되어 있는 모든 태그도 함께 삭제됩니다.

태그에 대한 자세한 내용은 AWS 리소스 태깅 및 태그 편집기 사용 설명서의 모범 사례 및 전략을 참조하세요.

> **Note**
>
> 태그는 기밀 또는 민감한 정보를 저장하기 위한 것이 아닙니다. 태그를 사용하여 이러한 유형의 데이터를 저장하지 마세요. 태그는 다른 AWS 서비스에서 액세스할 수 있습니다.

태그 추가

Amazon Inspector 리소스에 태그를 추가할 수 있습니다. 이러한 리소스에는 억제 규칙 및 CIS 스캔 구성이 포함됩니다. 태그를 사용하면 특정 기준에 따라 AWS 리소스를 분류할 수 있습니다. 이 주제에서는 Amazon Inspector 리소스에 태그를 추가하는 방법을 설명합니다.

Amazon Inspector 리소스에 태그 추가

억제 규칙 및 CIS 스캔 구성에 태그를 지정할 수 있습니다. 다음 절차에서는 콘솔과 Amazon Inspector API를 사용하여 태그를 추가하는 방법을 설명합니다.

콘솔에서 태그 추가

콘솔에서 Amazon Inspector 리소스에 태그를 추가할 수 있습니다.

억제 규칙에 태그 추가

생성 중에 억제 규칙에 태그를 추가할 수 있습니다. 자세한 내용은 억제 규칙 생성을 참조하세요.

억제 규칙을 편집하여 태그를 포함할 수도 있습니다. 자세한 내용은 억제 규칙 편집을 참조하세요.

CIS 스캔 구성에 태그 추가

생성 중에 CIS 스캔 구성에 태그를 추가할 수 있습니다. 자세한 내용은 CIS 스캔 구성 생성을 참조하세요.

태그를 포함하도록 CIS 스캔 구성을 편집할 수도 있습니다. 자세한 내용은 CIS 스캔 구성 편집을 참조하세요.

Amazon Inspector API를 사용하여 태그 추가

Amazon Inspector API를 사용하여 Amazon Inspector 리소스에 태그를 추가할 수 있습니다.

Amazon Inspector 리소스에 태그 추가

`TagResource` API를 사용하여 Amazon Inspector 리소스에 태그를 추가합니다. 명령에 리소스의 ARN과 태그의 키-값 페어를 포함해야 합니다. 다음 예제 명령은 억제 필터에 빈 리소스 ARN을 사용합니다. 키는 `CostAllocation`이고 값은 `dev`입니다. Amazon Inspector의 리소스 유형에 대한 자세한 내용은 서비스 승인 참조에서 Amazon Inspector2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

```
aws inspector2 tag-resource \
--resource-arn "arn:${Partition}:inspector2:${Region}:${Account}:owner/${OwnerId}/
filter/${FilterId}" \
```

```
--tags CostAllocation=dev \
--region us-west-2
```

생성 중 억제 규칙에 태그 추가

CreateFilter API를 사용하여 생성 중에 억제 규칙에 태그를 추가합니다.

```
aws inspector2 create-filter \
--name "ExampleSuppressionRuleECR" \
--action SUPPRESS \
--filter-criteria 'resourceType=[{comparison="EQUALS", value="AWS_ECR_IMAGE"}]' \
--tags Owner=ApplicationSecurity \
--region us-west-2
```

CIS 스캔 구성에 태그 추가

CreateCisScanConfiguration API를 사용하여 CIS 스캔 구성에 태그를 추가합니다.

```
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
```

태그 제거

Amazon Inspector 리소스에서 태그를 제거할 수 있습니다. 이러한 리소스에는 억제 규칙 및 CIS 스캔 구성이 포함됩니다. 태그를 사용하면 특정 기준에 따라 AWS 리소스를 분류할 수 있습니다. 이 주제에서는 Amazon Inspector 리소스에서 태그를 제거하는 방법을 설명합니다.

Amazon Inspector 리소스에서 태그 제거

억제 규칙 및 CIS 스캔 구성에서 태그를 제거할 수 있습니다. 다음 절차에서는 콘솔 및 Amazon Inspector API를 사용하여 태그를 제거하는 방법을 설명합니다.

콘솔에서 태그 제거

콘솔에서 Amazon Inspector 리소스에서 태그를 제거할 수 있습니다.

억제 규칙에서 태그 제거

더 이상 태그를 포함하지 않도록 억제 규칙을 편집하여 억제 규칙에서 태그를 제거할 수 있습니다. 자세한 내용은 억제 규칙 편집을 참조하세요.

CIS 스캔 구성에서 태그 제거

더 이상 태그를 포함하지 않도록 CIS 스캔 구성을 편집하여 CIS 스캔 구성에서 태그를 제거할 수 있습니다. 자세한 내용은 CIS 스캔 구성 편집을 참조하세요.

Amazon Inspector API를 사용하여 태그 제거

Amazon Inspector API를 사용하여 Amazon Inspector 리소스에서 태그를 제거할 수 있습니다.

Amazon Inspector 리소스에서 태그 제거

`UntagResource` API를 사용하여 Amazon Inspector 리소스에서 태그를 제거합니다.

다음 코드 조각은 `UntagResource`를 사용하여 Amazon Inspector 리소스에서 태그를 제거하는 방법의 예를 보여줍니다. 명령에 태그에 대한 리소스 및 키의 ARN을 포함해야 합니다. 다음 예제에서는 억제 필터에 빈 리소스 ARN을 사용합니다. 키는 `CostAllocation`입니다. Amazon Inspector의 리소스 유형에 대한 자세한 내용은 서비스 승인 참조에서 Amazon Inspector2에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

```
aws inspector2 untag-resource \
--resource-arn "arn:${Partition}:inspector2:${Region}:${Account}:owner/${OwnerId}/cis-configuration/${CISScanConfigurationId}" \
--tag-keys CostAllocation \
--region us-west-2
```

Amazon Inspector에서 사용량 및 비용 모니터링

Amazon Inspector 콘솔 및 API를 사용하여 사용자 환경에 대한 월별 Amazon Inspector 비용을 예측할 수 있습니다. 다중 계정 환경의 Amazon Inspector 관리자인 경우 환경의 총 비용과 모든 멤버 계정의 비용 지표를 확인할 수 있습니다. 이 단원에서는 사용량 통계에 액세스하고 사용 비용을 계산하는 방법에 대해 설명합니다.

사용량 콘솔 사용

콘솔에서 Amazon Inspector의 사용량과 예상 비용을 평가할 수 있습니다.

사용량 통계에 액세스하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 페이지 오른쪽 상단의 AWS 리전 선택기를 사용하여 비용을 모니터링할 리전을 선택합니다.
3. 탐색 창에서 사용량을 선택합니다.

계정별 탭의 계정 사용량에는 30일 기간을 기준으로 예상되는 총 비용이 표시됩니다. 예상 비용 열 아래의 테이블에서 값을 선택하면 해당 계정의 스캔 유형별 사용 내역을 확인할 수 있습니다. 이 세부 정보 창에서는 해당 계정에 대해 무료 평가판이 활성화된 스캔 유형도 확인할 수 있습니다.

조직의 위임 관리자인 경우 조직 내 각 계정에 대한 행이 테이블에 표시됩니다. 조직의 계정 연결이 해제된 경우 콘솔에 예상 비용이 -로 표시됩니다.

스캔 유형별 탭에서는 현재 30일 동안의 실제 사용량 내역을 스캔 유형별로 확인할 수 있습니다. 이는 계정별 탭에서 예상 비용을 계산하는 데 사용되는 정보입니다.

조직의 위임 관리자인 경우 조직 내 각 계정에 대한 사용량을 확인할 수 있습니다.

이 탭에서 다음 창 중 하나를 확장하여 사용량 통계를 확인할 수 있습니다.

Amazon EC2 스캔

Amazon Inspector 사용량 콘솔은 에이전트 기반 스캔 및 에이전트 없는 스캔에 대해 다음과 같은 지표를 추적합니다.

- 인스턴스 (평균) - Amazon Inspector는 적용 범위 시간을 사용하여 EC2 인스턴스 스캔에 필요한 평균 리소스 수를 계산합니다. 평균은 총 적용 범위 시간을 720시간(30일 기간의 시간 수)으로 나눈 값입니다.
- 적용 범위 시간 - Amazon EC2 스캔의 경우, 지난 30일 동안 Amazon Inspector가 계정 내 각 EC2 인스턴스에 대해 활성 적용 범위를 제공한 총 시간입니다. EC2 인스턴스의 경우, 적용 범위 시간은 Amazon Inspector에서 인스턴스를 발견한 시점부터 인스턴스가 종료 또는 중지되거나 태그에 의해 스캔에서 제외될 때까지의 시간입니다. 중지된 인스턴스를 다시 시작하거나 제외 태그를 제거하면 Amazon Inspector에서 적용을 재개하고 해당 인스턴스에 대한 적용 시간은 계속 누적됩니다.

CIS 인스턴스 스캔 - 계정의 인스턴스에 대해 수행된 총 CIS 스캔 횟수입니다.

Amazon ECR 스캔

최초 스캔 - 지난 30일 동안 계정의 이미지를 처음 스캔한 총 횟수입니다.

재스캔 - 지난 30일 동안 계정의 이미지를 재스캔한 총 횟수입니다. 재스캔은 Amazon Inspector에서 이전에 스캔한 ECR 이미지에 대해 수행되는 스캔입니다. ECR 리포지토리를 연속 스캔하도록 구성한 경우 Amazon Inspector에서 새로운 일반적인 취약성 및 노출(CVE)을 데이터베이스에 추가하면 자동으로 재스캔이 수행됩니다.

Lambda 스캔

Amazon Inspector 사용량 콘솔은 Lambda 표준 스캔 및 Lambda 코드 스캔에 대해 다음과 같은 지표를 추적합니다.

- Lambda 함수 수(평균) - Amazon Inspector는 적용 범위 시간을 사용하여 Lambda 함수 스캔에 필요한 평균 함수 수를 계산합니다. 평균은 총 적용 범위 시간을 720시간(30일 기간의 시간 수)으로 나눈 값입니다.
- 적용 범위 시간 - Lambda 함수 스캔의 경우, 지난 30일 동안 Amazon Inspector가 계정 내 각 Lambda 함수에 대해 활성 적용 범위를 제공한 총 시간입니다. AWS Lambda 함수의 경우 적용 범위 시간은 Amazon Inspector에서 함수를 발견한 시점부터 함수가 삭제되거나 검사에서 제외되는 시점까지 계산됩니다. 제외된 함수가 다시 포함되면 해당 함수에 대한 적용 범위 시간이 계속 누적됩니다.

Amazon Inspector의 사용 비용 계산 방식 이해

Amazon Inspector에서 제공하는 비용은 실제 비용이 아닌 예상 비용이므로 AWS Billing 콘솔의 비용과 다를 수 있습니다.

사용량 페이지에서 Amazon Inspector가 비용을 계산하는 방법에 대해서는 다음 사항을 참고하세요.

- 사용 비용은 현재 리전에만 적용됩니다. 스캔 유형당 가격은 AWS 리전에 따라 다릅니다. 리전당 정확한 가격을 검토하려면 Amazon Inspector 요금을 참조하세요.
- 모든 사용량 예상치는 으로 가장 가까운 금액(미국 달러 기준)으로 반올림됩니다.
- 할인은 예상 비용에 포함되지 않습니다.
- 예상 비용은 스캔 유형별 30일 사용 기간 동안의 총 비용을 나타냅니다. 계정 사용 기간이 30일 미만인 경우, Amazon Inspector는 현재 적용되는 리소스가 남은 30일 동안 계속 적용될 것으로 간주하여 30일 이후의 비용을 예상합니다.
- 스캔 유형별 비용은 다음을 기준으로 계산됩니다.
  - EC2 스캔: 비용에는 지난 30일 동안 Amazon Inspector에서 적용한 평균 EC2 인스턴스 수가 반영됩니다.
  - ECR 컨테이너 스캔: 비용에는 지난 30일 동안의 최초 이미지 스캔 횟수 + 이미지 재스캔 횟수의 합이 반영됩니다.
  - Lambda 표준 스캔: 비용에는 지난 30일 동안 Amazon Inspector에서 적용한 평균 Lambda 함수 수가 반영됩니다.
  - Lambda 코드 스캔: 비용에는 지난 30일 동안 Amazon Inspector에서 적용한 평균 Lambda 함수 수가 반영됩니다.

Amazon Inspector 무료 평가판 정보

Amazon Inspector의 각 스캔 유형에는 무료 평가판이 있습니다. 스캔 유형을 활성화하면 해당 스캔 유형에 대한 15일 무료 평가판에 자동으로 등록됩니다. 무료 평가판이 시작되면 스캔 유형을 비활성화하더라도 15일 후에 자동으로 만료됩니다.

> **Note**
>
> CIS 스캔에는 무료 평가판이 적용되지 않습니다.

Amazon Inspector의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. Amazon Inspector에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스규정 준수 프로그램 .
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon Inspector를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Amazon Inspector를 구성하는 방법을 보여줍니다. 또한 Amazon Inspector 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- Amazon Inspector의 데이터 보호
- Amazon Inspector용 Identity and Access Management
- Amazon Inspector 모니터링
- Amazon Inspector의 규정 준수 검증
- Amazon Inspector의 복원성
- Amazon Inspector의 인프라 보안
- Amazon Inspector의 인시던트 대응
- 인터페이스 엔드포인트(AWS PrivateLink)를 사용하여 Amazon Inspector에 액세스

Amazon Inspector의 데이터 보호

AWS 공동 책임 모델 Amazon Inspector의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 관한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon Inspector 또는 기타 AWS 서비스 에서 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

주제

- 저장된 데이터 암호화
- 전송 중 암호화

저장된 데이터 암호화

기본적으로 Amazon Inspector는 AWS 암호화 솔루션을 사용하여 저장 데이터를 저장합니다. Amazon Inspector에서는 다음과 같은 데이터를 암호화합니다.

- 로 수집된 리소스 인벤토리입니다 AWS Systems Manager.
- Amazon Elastic Container Registry 이미지에서 구문 분석된 리소스 인벤토리
- 의 AWS 소유 암호화 키를 사용하여 보안 조사 결과 생성 AWS Key Management Service

AWS 소유 키는 관리, 사용 또는 볼 수 없습니다. 그러나 데이터 암호화 키를 보호하기 위해 사용자가 별도의 조치를 취하거나 프로그램을 변경할 필요도 없습니다. 자세한 내용은 AWS 소유 키를 참조하세요.

Amazon Inspector를 비활성화하면 수집된 인벤토리 및 보안 조사 결과와 같이 저장 또는 유지 관리되는 모든 리소스가 영구적으로 삭제됩니다.

조사 결과 코드에 대한 저장 중 암호화

Amazon Inspector Lambda 코드 스캔의 경우, Amazon Inspector는 Amazon Q와 협력하여 코드에 취약성이 있는지 스캔합니다. 취약성이 탐지되면 Amazon Q는 취약성이 포함된 코드 스니펫을 추출하여 Amazon Inspector에서 액세스를 요청할 때까지 해당 코드를 저장합니다. 기본적으로 Amazon Q는 AWS 소유 키를 사용하여 추출된 코드를 암호화합니다. 그러나 암호화에 자체 고객 관리형 AWS KMS 키를 사용하도록 Amazon Inspector를 구성할 수 있습니다.

다음 워크플로우는 Amazon Inspector에서 사용자가 구성한 키를 사용하여 코드를 암호화하는 방법을 설명합니다.

1. Amazon Inspector UpdateEncryptionKey API를 사용하여 Amazon Inspector에 AWS KMS 키를 제공합니다.
2. Amazon Inspector는 AWS KMS 키에 대한 정보를 Amazon Q에 전달하고 Amazon Q는 나중에 사용할 수 있도록 정보를 저장합니다.
3. Amazon Q는 키 정책을 통해 Amazon Inspector에서 구성한 KMS 키를 사용합니다.
4. Amazon Q는 키에서 암호화된 데이터 AWS KMS 키를 생성하고 저장합니다. 이 데이터 키는 Amazon Q에 저장된 코드 데이터를 암호화하는 데 사용됩니다.
5. Amazon Inspector가 코드 스캔에서 데이터를 요청하면 Amazon Q는 KMS 키를 사용하여 데이터 키를 복호화합니다. Lambda 코드 스캔을 비활성화하면 Amazon Q가 연결된 데이터 키를 삭제합니다.

고객 관리형 키를 사용한 코드 암호화에 대한 권한

암호화의 경우 Amazon Inspector 및 Amazon Q가 다음 작업을 수행할 수 있도록 허용하는 문이 포함된 정책을 사용하여 KMS 키를 생성해야 합니다.

- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:Encrypt`
- `kms:GenerateDataKey`
- `kms:GenerateDataKeyWithoutPlainText`

정책 문

KMS 키를 생성할 때 다음 정책 설명을 사용할 수 있습니다.

> **Note**
>
> 를 12자리 AWS 계정 ID*account-id*로 바꿉니다. 를 Amazon Inspector 및 Lambda 코드 스캔을 활성화 AWS 리전 한 *Region*로 바꿉니다. *role-ARN*을 IAM 역할의 Amazon 리소스 이름으로 바꿉니다.

```
{
  "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"
    }
  }
}
```

정책 설명의 형식은 JSON입니다. 명령문을 포함한 후 정책을 검토하여 구문이 올바른지 확인합니다. 명령문이 정책의 마지막 명령문인 경우 이전 명령문의 닫는 괄호 뒤에 쉼표를 추가합니다. 명령문이 첫 번째 명령문이거나 정책의 기존 2개 명령문 사이에 있는 경우 명령문의 닫는 괄호 뒤에 쉼표를 추가합니다.

> **Note**
>
> Amazon Inspector는 패키지에서 추출된 코드 조각을 암호화하기 위한 권한 부여를 더 이상 지원하지 않습니다. 권한 부여 기반 정책을 사용하는 경우에도 조사 결과에 액세스할 수 있습니다. 그러나 KMS 키를 업데이트 또는 재설정하거나 Lambda 코드 스캔을 비활성화하는 경우 이 섹션에 설명된 KMS 키 정책을 사용해야 합니다.

계정의 암호화 키를 설정, 업데이트 또는 재설정하는 경우 AWS 관리형 정책과 같은 Amazon Inspector 관리자 정책을 사용해야 합니다`AmazonInspector2FullAccess`.

고객 관리형 키를 사용하여 암호화 구성

고객 관리형 키를 사용하여 계정 암호화를 구성하려면 고객 관리형 키를 사용한 코드 암호화에 대한 권한에 설명된 권한을 가진 Amazon Inspector 관리자여야 합니다. 또한 조사 결과와 동일한 AWS 리전에 있는 AWS KMS 키 또는 다중 리전 키가 필요합니다. 계정에서 기존 대칭 키를 사용하거나 AWS 관리 콘솔 또는 AWS KMS APIs. 자세한 내용은 AWS KMS 사용 설명서의 대칭 암호화 AWS KMS 키 생성을 참조하세요.

> **Note**
>
> 2025년 6월 13일부터 코드 조각 암호화/복호화 중에 CloudTrail에 기록된 AWS KMS 요청의 서비스 보안 주체가 "codeguru-reviewer"에서 "q"로 변경됩니다.

Amazon Inspector API를 사용하여 암호화 구성

암호화 키를 설정하려면 Amazon Inspector 관리자로 로그인한 상태에서 Amazon Inspector API의 UpdateEncryptionKey 작업을 수행합니다. API 요청에서 `kmsKeyId` 필드를 사용하여 사용하려는 AWS KMS 키의 ARN을 지정합니다. `scanType`에 `CODE`를 입력하고 `resourceType`에 `AWS_LAMBDA_FUNCTION`을 입력합니다.

UpdateEncryptionKey API를 사용하여 Amazon Inspector가 암호화에 사용하는 AWS KMS 키를 확인할 수 있습니다.

> **Note**
>
> 고객 관리형 키를 설정하지 않은 `GetEncryptionKey` 상태에서를 사용하려고 하면 작업이 `ResourceNotFoundException` 오류를 반환합니다. 이는 AWS 소유 키가 암호화에 사용되고 있음을 의미합니다.

Amazon Inspector 또는 Amazon Q에 대한 액세스를 거부하기 위해 키를 삭제하거나 정책을 변경하면 코드 취약성 조사 결과에 액세스할 수 없게 되며 계정에 대한 Lambda 코드 스캔이 실패합니다.

`ResetEncryptionKey`를 사용하여 AWS 소유 키를 사용하여 Amazon Inspector 조사 결과의 일부로 추출된 코드를 암호화할 수 있습니다.

전송 중 암호화

AWS 는 AWS 내부 시스템과 다른 AWS 서비스 간에 전송 중인 모든 데이터를 암호화합니다.는 평가를 위해 TLS(전송 계층 보안) 보호 채널을 AWS 통해 전송하는 고객 소유 EC2 인스턴스에서 원격 측정 데이터를 AWS Systems Manager 수집합니다. Security Hub CSPM으로 전송되는 Amazon ECR 및 AWS Lambda 함수 스캔 조사 결과는 TLS 보호 채널을 사용하여 암호화됩니다. 자세한 내용은 Systems Manager의 데이터 보호를 참조하여 SSM의 전송 중 데이터 암호화 방식을 이해하세요.

Amazon Inspector용 Identity and Access Management

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 인증(로그인) 및 권한 부여(권한 보유)를 통해 Amazon Inspector 리소스를 사용할 수 있는 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- Amazon Inspector에서 IAM을 사용하는 방법
- Amazon Inspector의 자격 증명 기반 정책 예
- AWS Amazon Inspector에 대한 관리형 정책
- Amazon Inspector에 서비스 연결 역할 사용
- Amazon Inspector 자격 증명 및 액세스 문제 해결

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청(참조Amazon Inspector 자격 증명 및 액세스 문제 해결)
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(Amazon Inspector에서 IAM을 사용하는 방법 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(Amazon Inspector의 자격 증명 기반 정책 예 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여에 Directory Service 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

Amazon Inspector에서 IAM을 사용하는 방법

IAM을 사용하여 Amazon Inspector에 대한 액세스를 관리하기 전에 Amazon Inspector에서 사용할 수 있는 IAM 특성에 대해 알아봅니다.

Amazon Inspector에서 사용할 수 있는 IAM 특성

IAM 특성	Amazon Inspector 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책 내 태그)	부분적
임시 자격 증명	예

IAM 특성	Amazon Inspector 지원
엔터티 권한	예
서비스 역할	아니요
서비스 연결 역할	예

Amazon Inspector 및 기타에서 대부분의 IAM 기능을 AWS 서비스 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 AWS 서비스 IAM으로 작업하는 섹션을 참조하세요.

Amazon Inspector의 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

Amazon Inspector의 자격 증명 기반 정책 예

Amazon Inspector 자격 증명 기반 정책 예제를 보려면 Amazon Inspector의 자격 증명 기반 정책 예 섹션을 참조하세요.

Amazon Inspector 내의 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM에서 교차 계정 리소스 액세스를 참조하세요.

Amazon Inspector에 대한 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

Amazon Inspector 작업 목록을 보려면 서비스 승인 참조에서 Amazon Inspector에서 정의한 작업을 참조하세요.

Amazon Inspector의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
inspector2
```

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

```
"Action": [
      "inspector2:action1",
      "inspector2:action2"
         ]
```

Amazon Inspector 자격 증명 기반 정책 예제를 보려면 Amazon Inspector의 자격 증명 기반 정책 예 섹션을 참조하세요.

Amazon Inspector에 대한 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

Amazon Inspector리소스 유형 및 해당 ARN의 목록을 보려면 서비스 승인 참조에서 Amazon Inspector에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 Amazon Inspector에서 정의한 작업을 참조하세요.

Amazon Inspector 자격 증명 기반 정책 예제를 보려면 Amazon Inspector의 자격 증명 기반 정책 예 섹션을 참조하세요.

Amazon Inspector에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

Amazon Inspector 조건 키 목록을 보려면 서비스 승인 참조에서 Amazon Inspector에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Amazon Inspector에서 정의한 작업을 참조하세요.

Amazon Inspector 자격 증명 기반 정책 예제를 보려면 Amazon Inspector의 자격 증명 기반 정책 예 섹션을 참조하세요.

Amazon Inspector의 ACL

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon Inspector를 사용한 ABAC

ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

Amazon Inspector에서 임시 보안 인증 정보 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는 AWS 서비스 섹션을 참조하세요.

Amazon Inspector에 대한 교차 서비스 위탁자 권한

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

Amazon Inspector의 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 Amazon Inspector 기능이 중단될 수 있습니다. Amazon Inspector에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

Amazon Inspector의 서비스 연결 역할

서비스 연결 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스 섹션을 참조하세요. 서비스 연결 역할 열에서 Yes가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

Amazon Inspector의 자격 증명 기반 정책 예

기본적으로 사용자 및 역할은 Amazon Inspector 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 Amazon Inspector에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조에서 Amazon Inspector에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- Amazon Inspector 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- 모든 Amazon Inspector 리소스에 대한 읽기 전용 액세스 허용
- 모든 Amazon Inspector 리소스에 대한 전체 액세스 허용

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 Amazon Inspector 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

Amazon Inspector 콘솔 사용

Amazon Inspector 콘솔에 액세스하려면 최소한의 권한이 있어야 합니다. 이러한 권한은 AWS 계정에서 Amazon Inspector 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 여전히 Amazon Inspector 콘솔을 사용할 수 있도록 하려면 Amazon Inspector *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "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": "*"
        }
```

```
    ]
}
```

모든 Amazon Inspector 리소스에 대한 읽기 전용 액세스 허용

이 예제에서는 모든 Amazon Inspector 리소스에 대한 읽기 전용 액세스를 허용하는 정책을 보여줍니다.

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": "*"
        }
    ]
}
```

모든 Amazon Inspector 리소스에 대한 전체 액세스 허용

이 예제에서는 모든 Amazon Inspector 리소스에 대한 전체 액세스를 허용하는 정책을 보여줍니다.

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 Amazon Inspector에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 줍니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AmazonInspector2FullAccess_v2

`AmazonInspector2FullAccess_v2` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 Amazon Inspector에 대한 전체 액세스 권한과 다른 관련된 서비스에 대한 액세스 권한을 부여합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `inspector2` - Amazon Inspector API에 대한 전체 액세스를 허용합니다.
- `codeguru-security` - 관리자가 계정의 보안 조사 결과 및 구성 설정을 검색할 수 있습니다.
- `iam` - Amazon Inspector가 서비스 연결 역할인 `AWSServiceRoleForAmazonInspector2` 및 `AWSServiceRoleForAmazonInspector2Agentless`을 생성할 수 있도록 허용합니다. Amazon Inspector에서 Amazon EC2 인스턴스, Amazon ECR 리포지토리 및 Amazon ECR 컨테이너 이미지에 대한 정보를 검색하는 등의 작업을 수행하려면 `AWSServiceRoleForAmazonInspector2`가 필요합니다. 또한 AWS KMS 키로 암호화된 Amazon EBS 스냅샷을 복호화해야 합니다. 자세한 내용은 Amazon Inspector에 서비스 연결 역할 사용 단원을 참조하십시오.
- `organizations` - 서비스 보안 주체만에 대한 서비스 연결 역할을 생성하고 AWS 계정,를 조직의 위임된 관리자 AWS 계정 로 등록하고, 조직의 위임된 관리자를 나열할 `AllowServicePrincipalBasedAccessToOrganizationApis` 수 있습니

다. `AllowOrganizationalBasedAccessToOrganizationApis`를 사용하면 정책 소유자가 조직 단위에 대한 정보, 특히 리소스 수준 ARNs을 검색할 수 있습니다.를 `AllowAccountsBasedAccessToOrganizationApis` 사용하면 정책 소유자가에 대한 정보, 특히 리소스 수준 ARNs을 검색할 수 있습니다 AWS 계정.를 `AllowAccessToOrganizationApis` 사용하면 정책 소유자가 조직 및 조직 정보와 AWS 서비스 통합된를 볼 수 있습니다. 이 정책을 사용하면 Inspector 정책 유형별로 필터링하고, 관리 계정에서 설정한 위임 리소스 정책을 보고, 계정에 적용된 유효한 Inspector 정책을 볼 수 있는 Inspector 조직 정책을 나열할 수 있습니다.

> **Note**
>
> Amazon Inspector는 더 이상 CodeGuru를 사용하여 Lambda 스캔을 수행하지 않습니다. AWS 는 2025년 11월 20일에 CodeGuru에 대한 지원을 중단할 예정입니다. 자세한 내용은 CodeGuru 보안에 대한 지원 종료를 참조하세요. Amazon Inspector는 이제 Amazon Q를 사용하여 Lambda 스캔을 수행하며 이 섹션에 설명된 권한이 필요하지 않습니다.

이 정책의 권한을 검토하려면 AWS 관리형 정책 참조 안내서의 AmazonInspector2FullAccess_v2를 참조하세요.

AWS 관리형 정책: AWSInspector2OrganizationsAccess

`AWSInspector2OrganizationsAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은의 조직에 대해 Amazon Inspector를 활성화하고 관리할 수 있는 관리 권한을 부여합니다 AWS Organizations. 이 정책에 대한 권한을 통해 조직 관리 계정은 Amazon Inspector의 위임된 관리자 계정을 지정할 수 있습니다. 또한, 위임된 관리자 계정을 통해 조직 계정을 멤버 계정으로 활성화할 수 있습니다.

이 정책은에 대한 권한만 제공합니다 AWS Organizations. 조직 관리 계정 및 위임된 관리자 계정에도 관련 작업에 대한 권한이 필요합니다. `AmazonInspector2FullAccess_v2` 관리형 정책을 사용하여 이러한 권한을 부여할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `organizations:ListAccounts` – 보안 주체가 조직의 일부인 계정 목록을 검색할 수 있습니다.
- `organizations:DescribeOrganization` – 보안 주체가 조직에 대한 정보를 검색할 수 있습니다.

- organizations:ListRoots – 보안 주체가 조직의 루트를 나열할 수 있습니다.
- organizations:ListDelegatedAdministrators - 보안 주체가 조직의 위임된 관리자를 나열할 수 있습니다.
- organizations:ListAWSServiceAccessForOrganization - 보안 주체 AWS 서비스 가 조직에서 사용하는를 나열할 수 있도록 허용합니다.
- organizations:ListOrganizationalUnitsForParent – 보안 주체가 상위 OU의 하위 조직 단위(OU)를 나열할 수 있습니다.
- organizations:ListAccountsForParent – 보안 주체가 상위 OU의 하위 계정을 나열할 수 있습니다.
- organizations:ListParents - 지정된 하위 조직 단위(OU) 또는 계정의 직속 상위 역할을 하는 루트 또는 OU를 나열합니다.
- organizations:DescribeAccount – 보안 주체가 조직의 계정에 대한 정보를 검색할 수 있습니다.
- organizations:DescribeOrganizationalUnit – 보안 주체가 조직의 OU에 대한 정보를 검색할 수 있습니다.
- organizations:ListPolicies - 지정된 유형의 조직에 있는 모든 정책의 목록을 검색합니다.
- organizations:ListPoliciesForTarget - 지정된 대상 루트, 조직 단위(OU) 또는 계정에 직접 연결된 정책을 나열합니다.
- organizations:ListTargetsForPolicy - 지정된 정책이 연결된 모든 루트, 조직 단위(OU) 및 계정을 나열합니다.
- organizations:DescribeResourcePolicy - 리소스 정책에 대한 정보를 검색합니다.
- organizations:EnableAWSServiceAccess – 위탁자가 Organizations를 통합할 수 있도록 허용합니다.
- organizations:RegisterDelegatedAdministrator – 위탁자가 위임된 관리자 계정을 지정할 수 있도록 허용합니다.
- organizations:DeregisterDelegatedAdministrator – 위탁자가 위임된 관리자 계정을 제거할 수 있도록 허용합니다.
- organizations:DescribePolicy - 정책에 대한 정보를 검색합니다.
- organizations:DescribeEffectivePolicy - 지정된 정책 유형 및 계정에 대한 유효 정책의 내용을 반환합니다.
- organizations:CreatePolicy - 루트, 조직 단위(OU) 또는 개인에 연결할 수 있는 지정된 유형의 정책을 생성합니다 AWS 계정.

- `organizations:UpdatePolicy` - 기존의 정책을 새로운 이름, 설명 또는 내용으로 업데이트합니다.
- `organizations:DeletePolicy` - 조직에서 지정된 정책을 삭제합니다.
- `organizations:AttachPolicy` - 정책을 루트, 조직 단위(OU) 또는 개인 계정에 연결합니다.
- `organizations:DetachPolicy` - 대상 루트, 조직 단위(OU) 또는 계정에서 정책을 분리합니다.
- `organizations:EnablePolicyType` - 루트에서 정책 유형을 활성화합니다.
- `organizations:DisablePolicyType` - 루트에서 조직 정책 유형을 비활성화합니다.
- `organizations:TagResource` - 지정된 리소스에 하나 이상의 태그를 추가합니다.
- `organizations:UntagResource` - 지정된 리소스에서 지정된 키가 있는 태그를 모두 제거합니다.
- `organizations:ListTagsForResource` - 지정된 리소스에 연결된 태그를 나열합니다.

이 정책의 권한을 검토하려면 AWS 관리형 정책 참조 안내서의 AWSInspector2OrganizationsAccess를 참조하세요.

AWS 관리형 정책: AmazonInspector2FullAccess

`AmazonInspector2FullAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 Amazon Inspector에 대한 전체 액세스를 허용하는 관리 권한을 부여합니다.

> **Important**
>
> 향상된 보안 및 Inspector 2 서비스 보안 주체에 대한 제한적인 권한을 얻으려면 AmazonInspector2FullAccess_v2를 사용하는 것이 좋습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `inspector2` - Amazon Inspector 기능에 대한 전체 액세스를 허용합니다.
- `iam` - Amazon Inspector가 서비스 연결 역할인 `AWSServiceRoleForAmazonInspector2` 및 `AWSServiceRoleForAmazonInspector2Agentless`을 생성할 수 있도록 허용합니다. Amazon

Inspector에서 Amazon EC2 인스턴스, Amazon ECR 리포지토리 및 컨테이너 이미지에 대한 정보를 검색하는 등의 작업을 수행하려면 `AWSServiceRoleForAmazonInspector2`가 필요합니다. 또한 Amazon Inspector가 VPC 네트워크를 분석하고 조직과 연결된 계정을 설명하는 데도 필요합니다. Amazon Inspector에서 Amazon EC2 인스턴스 및 Amazon EBS 스냅샷에 대한 정보를 검색하는 등의 작업을 수행하려면 `AWSServiceRoleForAmazonInspector2Agentless`가 필요합니다. 또한 AWS KMS 키로 암호화된 Amazon EBS 스냅샷을 복호화해야 합니다. 자세한 내용은 Amazon Inspector에 서비스 연결 역할 사용 단원을 참조하십시오.

- `organizations` - 관리자가 Amazon Inspector를 AWS Organizations의 조직에 사용할 수 있도록 허용합니다. 에서 Amazon Inspector에 대한 신뢰할 수 있는 액세스를 활성화하면 위임된 관리자 계정의 AWS Organizations구성원이 설정을 관리하고 조직 전체에서 결과를 볼 수 있습니다.
- `codeguru-security` - 관리자가 Amazon Inspector를 사용하여 정보 코드 스니펫을 검색하고 CodeGuru Security에서 저장한 코드의 암호화 설정을 변경할 수 있도록 허용합니다. 자세한 내용은 조사 결과 코드에 대한 저장 중 암호화 단원을 참조하십시오.

이 정책의 권한을 검토하려면 AWS 관리형 정책 참조 안내서의 AmazonInspector2FullAccess를 참조하세요.

AWS 관리형 정책: AmazonInspector2ReadOnlyAccess

`AmazonInspector2ReadOnlyAccess` 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 Amazon Inspector에 대한 읽기 전용 액세스를 허용하는 권한을 부여합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `inspector2` – Amazon Inspector 기능에 대한 읽기 전용 액세스를 허용합니다.
- `organizations` -의 조직에 대한 Amazon Inspector 적용 범위에 대한 세부 정보를 볼 수 AWS Organizations 있습니다. 또한 Inspector 정책 유형별로 필터링`ListPolicies`하여를 통해 Inspector 조직 정책을 보고,를 통해 위임 리소스 정책을 보고`DescribeResourcePolicy`,를 통해 계정에 적용된 유효한 Inspector 정책을 볼 수 있습니다`DescribeEffectivePolicy`. 이를 통해 사용자는 조직 정책을 수정하지 않고도 조직 정책을 통해 설정된 중앙 집중식 검사기 활성화를 이해할 수 있습니다.

- `codeguru-security` - CodeGuru Security에서 코드 스니펫을 검색할 수 있습니다. 또한 CodeGuru Security에 저장된 코드의 암호화 설정을 볼 수 있습니다.

이 정책의 권한을 검토하려면 AWS 관리형 정책 참조 안내서의 AmazonInspector2ReadOnlyAccess를 참조하세요.

AWS 관리형 정책: AmazonInspector2ManagedCisPolicy

`AmazonInspector2ManagedCisPolicy` 정책을 IAM 엔터티에 연결할 수 있습니다. 이 정책은 인스턴스의 CIS 스캔을 실행할 수 있도록 Amazon EC2 인스턴스에 권한을 부여하는 역할에 연결해야 합니다. IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이것은 EC2 인스턴스 내에 액세스 키를 저장하는 경우에 바람직한 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로필에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `inspector2` - CIS 스캔을 실행하는 데 사용되는 작업에 대한 액세스를 허용합니다.

이 정책의 권한을 검토하려면 AWS 관리형 정책 참조 안내서의 AmazonInspector2ManagedCisPolicy를 참조하세요.

AWS 관리형 정책: AmazonInspector2ServiceRolePolicy

`AmazonInspector2ServiceRolePolicy` 정책을 IAM 엔터티에 연결할 수 없습니다. 이 정책을 서비스 연결 역할에 연결하면 Amazon Inspector가 사용자를 대신하여 작업을 수행할 수 있습니다. 자세한 내용은 Amazon Inspector에 서비스 연결 역할 사용 단원을 참조하십시오.

AWS 관리형 정책: AmazonInspector2AgentlessServiceRolePolicy

`AmazonInspector2AgentlessServiceRolePolicy` 정책을 IAM 엔터티에 연결할 수 없습니다. 이 정책을 서비스 연결 역할에 연결하면 Amazon Inspector가 사용자를 대신하여 작업을 수행할 수 있습니다. 자세한 내용은 Amazon Inspector에 서비스 연결 역할 사용 단원을 참조하십시오.

AWS 관리형 정책: AmazonInspector2ManagedTelemetryPolicy

`AmazonInspector2ManagedTelemetryPolicy` 정책을 IAM 엔터티에 연결할 수 있습니다. 이 정책은 Amazon Inspector 원격 측정 작업에 대한 권한을 부여하여 서비스가 취약성 스캔을 위해 패키지 인벤토리 데이터를 수집하고 전송할 수 있도록 합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `inspector2-telemetry` - 패키지 조사 데이터 전송을 위한 작업에 대한 액세스를 허용합니다.

최신 버전의 JSON 정책 문서를 포함하여 정책에 대한 자세한 내용은 AWS 관리형 정책 참조 안내서의 AmazonInspector2ManagedTelemetryPolicy를 참조하세요.

AWS 관리형 정책에 대한 Amazon Inspector 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 Amazon Inspector의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 Amazon Inspector 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경	설명	Date
AWSInspector2OrganizationsAccess – 새 정책	Amazon Inspector는 정책을 통해 Amazon Inspector를 활성화하고 관리하는 데 필요한 권한을 부여하는 새로운 관리형 AWS Organizations 정책을 추가했습니다.	2026년 3월 3일
AmazonInspector2ManagedTelemetryPolicy – 새 정책	Amazon Inspector는 Amazon Inspector 원격 측정 작업에 대한 권한을 부여하는 새로운 관리형 정책을 추가하여 서비스가 취약성 스캔을 위해 패키지	2026년 2월 5일

변경	설명	Date
	인벤토리 데이터를 수집하고 전송할 수 있도록 합니다.	
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Amazon Inspector 연결성 분석을 위한 방화벽 메타데이터를 설명할 수 있는 새로운 권한을 추가했습니다. 또한 Amazon Inspector는 Amazon Inspector가 SSM 문서 와의 SSM 연결을 생성, 업데이트 및 시작할 수 있도록 추가 리소스 크기 조정을 추가했습니다`AWS-ConfigureAWSPackage` .	2026년 2월 3일
AmazonInspector2FullAccess_v2 및 AmazonInspector2ReadOnlyAccess – 기존 정책에 대한 업데이트	Amazon Inspector는 정책 소유자가 Inspector 조직 정책 및 위임 구성을 볼 수 있는 새 권한을 추가했습니다. 이를 통해 AWS Organizations 정책을 통한 Inspector 활성화의 중앙 집중식 관리 및 가시성을 지원합니다.	2025년 11월 14일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Amazon Inspector 정책이 Amazon Inspector AWS Organizations의 활성화 및 비활성화를 적용하도록 허용하는 새 권한을 추가했습니다 Amazon Inspector.	2025년 11월 10일
AmazonInspector2FullAccess_v2 – 새 정책	Amazon Inspector는 Amazon Inspector에 대한 전체 액세스 권한과 기타 관련 서비스에 대한 액세스를 제공하는 새로운 관리형 정책을 추가했습니다.	2025년 7월 3일

변경	설명	Date
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector가 IP 주소 및 인터넷 게이트웨이 Amazon Inspector를 설명할 수 있는 새 권한을 추가했습니다.	2025년 4월 29일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Amazon ECS 및 Amazon EKS 작업에 대한 읽기 전용 액세스를 허용하는 새 권한을 추가했습니다.	2025년 3월 25일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector가 AWS Lambda에서 함수 태그를 반환할 수 있는 새로운 권한이 추가되었습니다.	2024년 7월 31일
AmazonInspector2FullAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 서비스 연결 역할 `AWSServiceRoleForAmazonInspector2Agentless` 를 생성할 수 있는 권한이 추가되었습니다. 이를 통해 사용자는 Amazon Inspector를 활성화할 때 에이전트 기반 스캔 및 에이전트 없는 스캔을 수행할 수 있습니다.	2024년 4월 24일
AmazonInspector2ManagedCisPolicy – 새 정책	Amazon Inspector에는 인스턴스 프로파일의 일부로 사용하여 인스턴스에 대한 CIS 스캔을 허용할 수 있는 새로운 관리형 정책이 추가되었습니다.	2024년 1월 23일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector에서 대상 인스턴스에 대한 CIS 스캔을 시작할 수 있는 새로운 권한이 추가되었습니다.	2024년 1월 23일

변경	설명	Date
AmazonInspector2AgentlessServiceRolePolicy - 새 정책	에이전트 없는 EC2 인스턴스 스캔을 허용하도록 Amazon Inspector에 새 서비스 연결 역할 정책을 추가했습니다.	2023년 11월 27일
AmazonInspector2ReadOnlyAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 읽기 전용 사용자가 패키지 취약성 조사 결과에 대한 취약성 인텔리전스 세부 정보를 검색할 수 있는 새로운 권한을 추가했습니다.	2023년 9월 22일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector에서 Elastic Load Balancing 대상 그룹에 속하는 Amazon EC2 인스턴스의 네트워크 구성을 스캔할 수 있는 새로운 권한이 추가되었습니다.	2023년 8월 31일
AmazonInspector2ReadOnlyAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 읽기 전용 사용자가 리소스에 대한 Software Bill of Materials (SBOM)를 내보낼 수 있는 새로운 권한을 추가했습니다.	2023년 6월 29일
AmazonInspector2ReadOnlyAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 읽기 전용 사용자가 자신의 계정에 대한 Lambda 코드 스캔 조사 결과의 암호화 설정 세부 정보를 검색할 수 있는 새로운 권한을 추가했습니다.	2023년 6월 13일

변경	설명	Date
AmazonInspector2FullAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 사용자가 Lambda 코드 스캔 조사 결과의 코드를 암호화하도록 고객 관리형 KMS 키를 구성할 수 있는 새로운 권한을 추가했습니다.	2023년 6월 13일
AmazonInspector2ReadOnlyAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 읽기 전용 사용자가 자신의 계정에 대한 Lambda 코드 스캔 상태 및 조사 결과에 대한 세부 정보를 검색할 수 있는 새로운 권한을 추가했습니다.	2023년 5월 2일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Lambda 스캔을 활성화할 때 Amazon Inspector가 계정에서 AWS CloudTrail 서비스 연결 채널을 생성할 수 있는 새로운 권한을 추가했습니다. 이를 통해 Amazon Inspector는 사용자 계정의 CloudTrail 이벤트를 모니터링할 수 있습니다.	2023년 4월 30일
AmazonInspector2FullAccess – 기존 정책에 대한 업데이트	Amazon Inspector에서 사용자가 Lambda 코드 스캔의 코드 취약성 조사 결과에 대한 세부 정보를 검색할 수 있는 새로운 권한을 추가했습니다.	2023년 4월 21일

변경	설명	Date
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector에서 Amazon EC2 심층 검사를 위해 고객이 정의한 사용자 지정 경로에 대한 정보를 Amazon EC2 Systems Manager로 보낼 수 있는 새로운 권한이 추가되었습니다.	2023년 4월 17일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Lambda 스캔을 활성화할 때 Amazon Inspector가 계정에서 AWS CloudTrail 서비스 연결 채널을 생성할 수 있는 새로운 권한을 추가했습니다. 이를 통해 Amazon Inspector는 사용자 계정의 CloudTrail 이벤트를 모니터링할 수 있습니다.	2023년 4월 30일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Amazon Inspector가 AWS Lambda 함수의 개발자 코드 스캔을 요청하고 Amazon CodeGuru Security로부터 스캔 데이터를 수신할 수 있는 새로운 권한을 추가했습니다. 또한 Amazon Inspector에서 IAM 정책을 검토할 수 있는 권한이 추가되었습니다. Amazon Inspector는 이 정보를 사용하여 Lambda 함수의 코드 취약성을 스캔합니다.	2023년 2월 28일

변경	설명	Date
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 AWS Lambda 함수가 마지막으로 호출된 시간에 대한 정보를 CloudWatch에서 Amazon Inspector 검색할 수 있는 새 문을 추가했습니다. Amazon Inspector는 이 정보를 사용하여 사용자 환경에서 지난 90일 동안 활성화된 Lambda 함수에 초점을 맞추어 스캔을 수행합니다.	2023년 2월 20일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector는 Amazon Inspector가 각 AWS Lambda 함수와 연결된 각 계층 버전을 포함하여 함수에 대한 정보를 검색할 수 있는 새 문을 추가했습니다. Amazon Inspector는 이 정보를 사용하여 Lambda 함수의 보안 취약성을 스캔합니다.	2022년 11월 28일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector에서 SSM 연결 실행을 설명할 수 있는 새로운 작업이 추가되었습니다. 또한 Amazon Inspector에서 `AmazonInspector2` 소유 SSM 문서와 SSM 연결을 생성, 업데이트, 삭제 및 시작할 수 있도록 리소스 범위가 추가되었습니다.	2022년 8월 31일

변경	설명	Date
AmazonInspector2ServiceRolePolicy - 기존 정책에 대한 업데이트	Amazon Inspector는 Amazon Inspector가 다른 AWS 파티션에서 소프트웨어 인벤토리를 수집할 수 있도록 정책의 리소스 범위를 업데이트했습니다.	2022년 8월 12일
AmazonInspector2ServiceRolePolicy – 기존 정책에 대한 업데이트	Amazon Inspector에서 SSM 연결을 생성, 삭제 및 업데이트할 수 있도록 작업의 리소스 범위가 재구성되었습니다.	2022년 8월 10일
AmazonInspector2ReadOnlyAccess – 새 정책	Amazon Inspector 기능에 대한 읽기 전용 액세스를 허용하는 새로운 정책이 추가되었습니다.	2022년 1월 21일
AmazonInspector2FullAccess – 새 정책	Amazon Inspector 기능에 대한 전체 액세스를 허용하는 새로운 정책이 추가되었습니다.	2021년 11월 29일
AmazonInspector2ServiceRolePolicy – 새 정책	Amazon Inspector에서 사용자를 대신하여 다른 서비스의 작업을 수행할 수 있도록 허용하는 새로운 정책이 추가되었습니다.	2021년 11월 29일
Amazon Inspector에서 변경 사항 추적 시작	Amazon Inspector가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2021년 11월 29일

Amazon Inspector에 서비스 연결 역할 사용

Amazon Inspector는 라는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다`AWSServiceRoleForAmazonInspector2`. 이 서비스 연결 역할은 Amazon Inspector에 직접 연

결되는 IAM 역할입니다. Amazon Inspector에서 사전 정의하며 Amazon Inspector가 AWS 서비스 사용자를 대신하여 다른를 호출하는 데 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로 Amazon Inspector를 더 쉽게 설정할 수 있습니다. Amazon Inspector는 서비스 연결 역할의 권한을 정의하며, 다르게 정의되지 않는 한 Amazon Inspector만 해당 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

IAM 엔터티(예: 그룹 또는 역할)가 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요. 먼저 관련 리소스를 삭제해야만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스 액세스 권한을 실수로 삭제할 수 없기 때문에 Amazon Inspector 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 AWS IAM으로 작업하는 서비스를 참조하고 서비스 연결 역할 열에서 예인 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 검토하려면 예 링크를 선택합니다.

Amazon Inspector의 서비스 연결 역할 권한

Amazon Inspector는 `AWSServiceRoleForAmazonInspector2`라는 관리형 정책을 사용합니다. 이 서비스 연결 역할은 `inspector2.amazonaws.com` 서비스에 해당 역할을 맡깁니다.

역할에 대한 권한 정책(`AmazonInspector2ServiceRolePolicy`)을 통해 Amazon Inspector에서는 다음과 같은 작업을 수행할 수 있습니다.

- Amazon Elastic Compute Cloud(Amazon EC2) 작업을 사용하여 인스턴스 및 네트워크 경로에 대한 정보를 검색합니다.
- AWS Systems Manager 작업을 사용하여 Amazon EC2 인스턴스에서 인벤토리를 검색하고 사용자 지정 경로에서 타사 패키지에 대한 정보를 검색합니다.
- 작업을 사용하여 대상 인스턴스에 AWS Systems Manager `SendCommand` 대한 CIS 스캔을 호출합니다.
- Amazon Elastic Container Registry 작업을 사용하여 컨테이너 이미지에 대한 정보를 검색합니다.
- AWS Lambda 작업을 사용하여 Lambda 함수에 대한 정보를 검색합니다.
- AWS Organizations 작업을 사용하여 연결된 계정을 설명합니다.
- CloudWatch 작업을 사용하여 Lambda 함수가 마지막으로 간접적으로 호출된 시간에 대한 정보를 검색합니다.
- 일부 IAM 작업을 사용하여 Lambda 코드에 보안 취약성을 일으킬 수 있는 IAM 정책에 대한 정보를 검색합니다.

- Amazon Q 작업을 사용하여 Lambda 함수의 코드 스캔을 수행합니다. Amazon Inspector는 다음 Amazon Q 작업을 사용합니다.
  - codeguru-security:CreateScan – Amazon Q 스캔을 생성할 수 있는 권한을 부여합니다.
  - codeguru-security:GetScan – Amazon Q 스캔 메타데이터를 검색할 수 있는 권한을 부여합니다.
  - codeguru-security:ListFindings – Amazon Q에서 생성한 조사 결과를 검색할 수 있는 권한을 부여합니다.
  - codeguru-security:DeleteScansByCategory – Amazon Inspector에서 시작한 스캔을 삭제할 수 있는 권한을 Amazon Q에 부여합니다.
  - codeguru-security:BatchGetFindings – Amazon Q에서 생성한 특정 조사 결과를 일괄 검색할 수 있는 권한을 부여합니다.
- 일부 Elastic Load Balancing 작업을 사용하여 Elastic Load Balancing 대상 그룹에 속하는 EC2 인스턴스의 네트워크 스캔을 수행합니다.
- Amazon ECS 및 Amazon EKS 작업을 사용하여 클러스터 및 작업을 보고 작업을 설명할 수 있는 읽기 전용 액세스를 허용합니다.
- AWS Organizations 작업을 사용하여 조직 전체에서 Amazon Inspector의 위임된 관리자를 나열합니다.
- Amazon Inspector 작업을 사용하여 조직 전체에서 Amazon Inspector를 활성화 및 비활성화합니다.
- Amazon Inspector 작업을 사용하여 위임된 관리자 계정을 지정하고 조직 전체에서 멤버 계정을 연결합니다.

> **Note**
>
> Amazon Inspector는 더 이상 CodeGuru를 사용하여 Lambda 스캔을 수행하지 않습니다. AWS 는 2025년 11월 20일에 CodeGuru에 대한 지원을 중단할 예정입니다. 자세한 내용은 CodeGuru 보안에 대한 지원 종료를 참조하세요. Amazon Inspector는 이제 Amazon Q를 사용하여 Lambda 스캔을 수행하며 이 섹션에 설명된 권한이 필요하지 않습니다.

이 정책에 대한 권한을 보려면 AWS 관리형 정책 참조 가이드에서 AmazonInspector2ServiceRolePolicy를 참조하세요.

Amazon Inspector의 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS API에서 Amazon Inspector를 활성화하면 Amazon Inspector가 서비스 연결 역할을 생성합니다.

Amazon Inspector의 서비스 연결 역할 편집

Amazon Inspector에서는 AWSServiceRoleForAmazonInspector2 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있으므로 역할 이름을 변경할 수 없습니다. 그러나 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

Amazon Inspector의 서비스 연결 역할 삭제

Amazon Inspector를 더 이상 사용하지 않을 경우에는 AWSServiceRoleForAmazonInspector2 서비스 연결 역할을 삭제하는 것이 좋습니다. 역할을 삭제하려면 먼저 역할 AWS 리전 이 활성화된 각에서 Amazon Inspector를 비활성화해야 합니다. Amazon Inspector를 비활성화해도 역할은 삭제되지 않습니다. 따라서 Amazon Inspector를 다시 활성화하면 기존 역할을 사용할 수 있습니다. 이렇게 하면 적극적으로 모니터링되거나 유지 관리되지 않는 미사용 개체를 방지할 수 있습니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

이 서비스 연결 역할을 삭제한 다음 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. Amazon Inspector를 활성화하면 Amazon Inspector에서 서비스 연결 역할을 자동으로 다시 생성합니다.

> **Note**
>
> 리소스를 삭제하려고 할 때 Amazon Inspector 서비스에서 해당 역할을 사용 중이면 삭제가 실패할 수 있습니다. 이 경우 몇 분 정도 기다렸다가 작업을 다시 시도하세요.

IAM 콘솔, AWS CLI또는 AWS API를 사용하여 AWSServiceRoleForAmazonInspector2 서비스 연결 역할을 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

Amazon Inspector 에이전트 없는 스캔을 위한 서비스 연결 역할 권한

Amazon Inspector 에이전트 없는 스캔은 AWSServiceRoleForAmazonInspector2Agentless라는 서비스 연결 역할을 사용합니다. 이 SLR을 사용하면 Amazon Inspector가 사용자 계정에서 Amazon EBS 볼륨 스냅샷을 생성한 다음 해당 스냅샷의 데이터에 액세스할 수 있습니다. 이 서비스 연결 역할은 agentless.inspector2.amazonaws.com 서비스에 해당 역할을 맡깁니다.

> **Important**
>
> 이 서비스 연결 역할의 문은 Amazon Inspector가 `InspectorEc2Exclusion` 태그를 사용하여 스캔에서 제외한 모든 EC2 인스턴스에 대해 에이전트 없는 스캔을 수행하는 것을 방지합니다. 또한 이 문은 암호화하는 데 사용된 KMS 키에 `InspectorEc2Exclusion` 태그가 있는 경우 Amazon Inspector가 볼륨의 암호화된 데이터에 액세스하는 것을 방지합니다. 자세한 내용은 Amazon Inspector 스캔에서 인스턴스 제외 단원을 참조하십시오.

역할에 대한 권한 정책(`AmazonInspector2AgentlessServiceRolePolicy`)을 통해 Amazon Inspector에서는 다음과 같은 작업을 수행할 수 있습니다.

- Amazon Elastic Compute Cloud(Amazon EC2) 작업을 사용하여 EC2 인스턴스, 볼륨 및 스냅샷에 관한 정보를 검색합니다.
  - Amazon EC2 태그 지정 작업을 사용하여 스캔을 위해 `InspectorScan` 태그 키로 스냅샷에 태그를 지정합니다.
  - Amazon EC2 스냅샷 작업을 사용하여 스냅샷을 생성하고, `InspectorScan` 태그 키로 스냅샷에 태그를 지정한 다음, `InspectorScan` 태그 키로 태그가 지정된 Amazon EBS 볼륨의 스냅샷을 삭제합니다.
- Amazon EBS 작업을 사용하면 `InspectorScan` 태그 키로 태그가 지정된 스냅샷에서 정보를 검색합니다.
- 선택 AWS KMS 복호화 작업을 사용하여 AWS KMS 고객 관리형 키로 암호화된 스냅샷을 복호화합니다. Amazon Inspector는 스냅샷을 암호화하는 데 사용된 KMS 키에 `InspectorEc2Exclusion` 태그가 지정된 경우 스냅샷을 복호화하지 않습니다.

역할은 다음과 같은 권한 정책으로 구성됩니다.

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/*"
  }
 ]
}
```

에이전트 없는 스캔을 위한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS API에서 Amazon Inspector를 활성화하면 Amazon Inspector가 서비스 연결 역할을 생성합니다.

에이전트 없는 스캔을 위한 서비스 연결 역할 편집

Amazon Inspector에서는 `AWSServiceRoleForAmazonInspector2Agentless` 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있으므로 역할 이름을 변경할 수 없습니다. 그러나 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

에이전트 없는 스캔을 위한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 특성 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔티티가 없도록 합니다.

> **Important**
>
> `AWSServiceRoleForAmazonInspector2Agentless` 역할을 삭제하려면 에이전트 없는 스캔을 사용할 수 있는 모든 리전에서 스캔 모드를 에이전트 기반으로 설정해야 합니다.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForAmazonInspector2Agentless 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서에서 서비스 연결 역할 삭제를 참조하세요.

Amazon Inspector 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Amazon Inspector 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- Amazon Inspector에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 Amazon Inspector 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

Amazon Inspector에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 `inspector2:`*`GetWidget`* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 inspector2:GetWidget on resource: my-example-widget
```

이 경우, `inspector2:`*`GetWidget`* 작업을 사용하여 *`my-example-widget`* 리소스에 액세스할 수 있도록 `mateojackson` 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon Inspector에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예제 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 Amazon Inspector에서 태스크를 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 Amazon Inspector 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- Amazon Inspector에서 이러한 특성을 지원하는지 여부를 알아보려면 Amazon Inspector에서 IAM을 사용하는 방법 섹션을 참조하세요.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

Amazon Inspector 모니터링

모니터링은 Amazon Inspector 및 기타 AWS 솔루션의 가용성, 안정성 및 성능을 유지하는 데 중요한 부분입니다.는 Amazon Inspector를 모니터링하고, 발생하는 문제를 보고하고, 이러한 문제를 해결하기 위한 조치를 취할 수 있는 도구를 AWS 제공합니다.

- Amazon EventBridge는 이벤트를 사용하여 애플리케이션 구성 요소를 함께 연결하는 AWS 서비스이므로 확장 가능한 이벤트 기반 애플리케이션을 더 쉽게 구축할 수 있습니다. EventBridge는 애플

리케이션, Software-as-a-Service(SaaS) 애플리케이션, AWS 서비스 및 경로에서 실시간 데이터 스트림을 제공하므로 서비스에서 발생하는 이벤트를 모니터링하고 이벤트 기반 아키텍처를 구축할 수 있습니다.

- AWS CloudTrail는에 의해 또는를 대신하여 수행된 API 호출 및 관련 이벤트를 캡처하는 AWS 서비스입니다 AWS 계정. CloudTrail은 사용자가 지정한 Amazon S3 버킷에 로그 파일을 전송하므로 호출한 사용자 및 계정 AWS, 호출이 수행된 소스 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다.

를 사용하여 Amazon Inspector API 호출 로깅 AWS CloudTrail

Amazon Inspector는 Amazon Inspector AWS 서비스에서 IAM 사용자 또는 역할 또는가 수행한 작업의 레코드를 제공하는 AWS CloudTrail서비스와 통합됩니다. CloudTrail은 Amazon Inspector에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 캡처되는 직접 호출에는 Amazon Inspector 콘솔로부터의 직접 호출과 Amazon Inspector API 작업에 대한 직접 호출이 포함됩니다. 추적을 생성하면 Amazon Inspector 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 다음을 확인할 수 있습니다.

- Amazon Inspector에 보낸 요청
- 요청이 발생한 IP 주소
- 요청한 사람
- 요청이 발생한 시간

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 Amazon Inspector 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. Amazon Inspector에서 활동이 발생하면 해당 활동은 이벤트 기록에서 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

Amazon Inspector에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 AWS 리전에 트레일이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한

CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취 AWS 서비스 하도록 다른를 구성할 수 있습니다. 자세한 정보는 다음의 주제를 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 계정에서 CloudTrail 로그 파일 수신
- 여러 리전에서 CloudTrail 로그 파일 수신

모든 Amazon Inspector 작업은 Amazon CloudTrail에서 로깅합니다. Amazon Inspector가 수행할 수 있는 모든 작업은 Amazon Inspector API 참조에 문서화되어 있습니다. 예를 들어 `CreateFindingsReport`, `ListCoverage` 및 `UpdateOrganizationConfiguration` 작업을 직접적으로 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 아니면 IAM 사용자 자격 증명으로 했는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 자격 증명을 사용하여 요청했는지 여부
- 다른 AWS 서비스에서 요청했는지 여부

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

Amazon Inspector 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터 단일 요청을 나타냅니다. 이벤트에는 요청된 작업, 작업 날짜 및 시간, 요청 파라미터 등에 대한 정보가 포함되어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

CloudTrail의 Amazon Inspector 스캔 정보

Amazon Inspector 스캔은 CloudTrail과 통합됩니다. 모든 Amazon Inspector 스캔 API 작업은 관리 이벤트로 로깅됩니다. Amazon Inspector가 CloudTrail에 로깅하는 Amazon Inspector 스캔 API 작업의 목록은 Amazon Inspector API 참조의 Amazon Inspector 스캔을 참조하세요.

다음은 ScanSbom 태스크를 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.


```
{
    "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"
}
```


Amazon Inspector의 규정 준수 검증

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 내를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 AWS 서비스참조하세요.

Amazon Inspector의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 많으며 중복성이 높은 네트워킹에 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

Amazon Inspector의 인프라 보안

관리형 서비스인 Amazon Inspector는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon Inspector에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

Amazon Inspector의 인시던트 대응

AWS에서는 보안을 가장 중요하게 생각합니다. 공동 AWS 책임 모델에서 "클라우드 보안"에 언급된 대로 AWS 는 AWS 클라우드에서 모든 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 는 Amazon Inspector 서비스와 관련된 모든 인시던트 대응도 책임집니다.

AWS 고객은 AWS 클라우드에서 보안을 유지할 책임이 있습니다. 즉, 액세스한 모든 AWS 도구와 기능을 포함하여 구현하도록 선택한 보안을 제어할 수 있습니다. 또한 공동 책임 모델에 따라 인시던트 대응에 대한 책임은 사용자 측에 있다는 의미이기도 합니다.

AWS 클라우드에서 실행되는 애플리케이션의 모든 목표를 충족하는 보안 기준을 설정하면 대응할 수 있는 편차를 감지할 수 있습니다. 인시던트 대응은 복잡한 주제이므로 인시던트 대응의 영향과 선택한 사항이 기업 목표에 어떤 영향을 미칠 수 있는지 더 잘 이해하려면 AWS 보안 인시던트 대응 가이드, AWS 보안 모범 사례, AWS 클라우드 채택 프레임워크: 보안 관점 리소스를 검토하세요.

인터페이스 엔드포인트(AWS PrivateLink)를 사용하여 Amazon Inspector에 액세스

AWS PrivateLink 를 사용하여 VPC와 Amazon Inspector 간에 프라이빗 연결을 생성할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 Amazon Inspector에 액세스할 수 있습니다. VPC의 인스턴스에서 Amazon Inspector에 액세스하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 Amazon Inspector로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은 AWS PrivateLink 가이드의를 AWS 서비스 통한 액세스를 AWS PrivateLink 참조하세요.

Amazon Inspector 고려 사항

Amazon Inspector에 대한 인터페이스 엔드포인트를 설정하려면 먼저 AWS PrivateLink 가이드의 고려 사항을 검토합니다.

Amazon Inspector는 인터페이스 엔드포인트를 통해 모든 API 작업에 대한 직접 호출을 지원합니다.

Amazon Inspector에는 VPC 엔드포인트 정책이 지원되지 않습니다. 기본적으로 인터페이스 엔드포인트를 통해 Amazon Inspector에 대한 전체 액세스가 허용됩니다. 또는 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 인터페이스 엔드포인트를 통해 Amazon Inspector로 향하는 트래픽을 제어할 수 있습니다.

Amazon Inspector용 인터페이스 엔드포인트 생성

Amazon VPC 콘솔 또는 ()를 사용하여 Amazon Inspector에 대한 인터페이스 엔드포인트를 생성할 수 있습니다AWS CLI. AWS Command Line Interface 자세한 내용은 AWS PrivateLink 안내서의 인터페이스 엔드포인트 생성을 참조하세요.

Amazon Inspector의 인터페이스 엔드포인트를 생성할 경우 다음과 같은 서비스 이름 중 하나를 사용합니다.

```
com.amazonaws.region.inspector2
```

```
com.amazonaws.region.inspector-scan
```

*region*을 해당의 AWS 리전 코드로 바꿉니다 AWS 리전.

인터페이스 엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 기본 리전 DNS 이름(예: 미국 동부(버지니아 북부)의 경우 `service-name.us-east-1.amazonaws.com` 또는 `service-name.us-east-1.api.aws.com`)을 사용하여 Amazon Inspector에 API 요청을 할 수 있습니다.

Amazon Inspector 통합

Amazon Inspector는 다른 AWS 서비스와 통합됩니다. 이러한 서비스는 Amazon Inspector에서 데이터를 수집할 수 있으므로 다양한 방식으로 조사 결과를 확인할 수 있습니다. 자세히 알아보려면 다음 통합 옵션을 검토하세요.

에서 Amazon Inspector 사용 AWS Organizations

AWS Organizations는 AWS 환경을 중앙에서 관리하고 관리하는 데 도움이 됩니다. AWS Organizations 정책을 사용하여 조직의 여러 계정에서 Amazon Inspector를 자동으로 활성화하고 관리할 수 있습니다.

Amazon Inspector 조직 정책을 통해 다음을 수행할 수 있습니다.

- 조직 전체에서 Amazon Inspector 스캔 유형(EC2, ECR, Lambda, 코드 리포지토리)을 중앙에서 활성화
- 조직에 가입하는 새 계정에 Amazon Inspector 활성화 자동 적용
- 조직 단위 전반에 걸쳐 일관된 스캔 적용 범위 적용
- 멤버 계정이 필수 스캔을 비활성화하지 못하도록 방지

조직 정책은 리소스 유형 활성화를 제어하는 반면, 위임된 관리자는 스캔 구성 설정을 제어합니다. 조직 정책이 위임된 관리자 및 멤버 계정 권한과 상호 작용하는 방법에 대한 자세한 내용은 섹션을 참조하세요를 사용하여 Amazon Inspector에서 여러 계정 관리 AWS Organizations. Amazon Inspector 정책 생성에 대한 자세한 지침은 Amazon Inspector 정책 AWS Organizations 설명서를 참조하세요.

Amazon Inspector와 Amazon ECR 통합

Amazon Elastic Container Registry(Amazon ECR)는 프라이빗 레지스트리를 지원하는 AWS관리형 컨테이너 이미지 레지스트리입니다. Amazon ECR 프라이빗 레지스트리는 가용성 및 확장성이 뛰어난 아키텍처에서 컨테이너 이미지를 호스팅합니다. Amazon Inspector를 사용하여 Amazon ECR 리포지토리에 있는 컨테이너 이미지에서 취약한 운영 체제 패키지 및 프로그래밍 언어 패키지가 있는지 스캔할 수 있습니다. 자세한 내용은 Amazon Inspector와 Amazon Elastic Container Registry(Amazon ECR) 통합 단원을 참조하십시오.

와 Amazon Inspector 통합 AWS Security Hub CSPM

AWS Security Hub CSPM는의 보안 상태에 대한 포괄적인 보기를 AWS 제공하며 Security Hub CSPM이 AWS 계정, 서비스 및 지원되는 제품에서 보안 데이터를 수집하는 보안 업계 표준 및 모범 사례를 기준으로 환경을 확인하는 데 도움이 됩니다. Security Hub CSPM을 사용하여 Amazon Inspector 조사 결과 데이터를 수집하고 모든 통합 AWS 서비스 및 AWS 파트너 네트워크 제품의 조사 결과를 위한 중앙 위치를 생성할 수 있습니다. 자세한 내용은 와 Amazon Inspector 통합 AWS Security Hub CSPM 단원을 참조하십시오.

Amazon Inspector와 Amazon Elastic Container Registry(Amazon ECR) 통합

Amazon Elastic Container Registry는 Docker 및 OCI 이미지와 AWS 아티팩트를 지원하는 완전 관리형 컨테이너 레지스트리입니다. Amazon ECR을 사용하는 경우 컨테이너 레지스트리에 대해 고급 스캔을 활성화할 수 있습니다. Amazon ECR을 사용하는 경우, 레지스트리에 대한 고급 스캔을 활성화하여 Amazon Inspector에서 컨테이너 이미지를 자동으로 탐지하고 취약한 운영 체제 패키지 및 프로그래밍 언어 패키지가 있는지 스캔할 수 있습니다. 이 통합을 통해 컨테이너 이미지에 대한 Amazon Inspector 조사 결과를 확인하고 Amazon ECR 콘솔에서 스캔 빈도 및 범위를 관리할 수 있습니다. 자세한 내용은 Amazon Inspector로 Amazon ECR 컨테이너 이미지 스캔을 참조하세요.

통합 활성화

Amazon Inspector 콘솔 또는 API를 통해 Amazon Inspector 스캔을 활성화하거나, Amazon ECR 콘솔 또는 API를 통해 Amazon Inspector의 고급 스캔을 사용하도록 리포지토리를 구성하여 통합을 활성화할 수 있습니다.

Amazon Inspector를 통한 통합 활성화에 대한 자세한 내용은 Amazon Inspector의 자동 스캔 유형 섹션을 참조하세요.

Amazon ECR에서 고급 스캔을 활성화하고 구성하는 방법에 대한 자세한 내용은 Amazon ECR 사용 설명서에서 고급 스캔을 참조하세요.

다중 계정 환경과의 통합 사용

다중 계정 환경의 멤버인 경우 Amazon ECR을 통해 고급 검색을 활성화할 수 있습니다. 하지만 활성화한 후에는 Amazon Inspector 위임 관리자만 비활성화할 수 있습니다. 비활성화되면 기본 스캔으로 돌아갑니다. 자세한 내용은 Amazon Inspector 비활성화 단원을 참조하십시오.

와 Amazon Inspector 통합 AWS Security Hub CSPM

Security Hub CSPM은의 보안 상태에 대한 포괄적인 보기를 제공합니다 AWS. 이를 통해 고객 환경에서 보안 업계 표준 및 모범 사례를 준수하는지 확인할 수도 있습니다. Security Hub CSPM은 AWS 계정, 서비스 및 지원되는 제품에서 보안 데이터를 수집합니다. 이 정보를 사용하여 보안 추세를 분석하고 보안 문제를 식별할 수 있습니다. Security Hub CSPM과의 Amazon Inspector 통합을 활성화하면 Amazon Inspector가 조사 결과를 Security Hub CSPM으로 전송할 수 있으며 Security Hub CSPM은 보안 태세의 일부로 해당 조사 결과를 분석할 수 있습니다.

Security Hub CSPM은 보안 문제를 조사 결과로 추적합니다. 일부 결과는 다른 AWS 서비스 또는 타사 제품에서 감지된 보안 문제의 결과일 수 있습니다. Security Hub CSPM은 일련의 규칙을 사용하여 보안 문제를 감지하고 결과를 생성하며 도구를 제공하므로 결과를 관리할 수 있습니다. Security Hub CSPM은 Amazon Inspector에서 조사 결과가 닫히면 Amazon Inspector 조사 결과를 보관합니다. 또한 조사 결과의 기록과 조사 결과 세부 정보를 볼 수 있으며, 조사 결과에 대한 조사 상태를 추적할 수 있습니다.

Security Hub CSPM은 AWS Security Finding Format(ASFF)의 조사 결과를 처리합니다. 이 형식에는 고유 식별자, 심각도 수준, 영향을 받는 리소스, 문제 해결 지침, 워크플로 상태 및 컨텍스트 정보와 같은 세부 정보가 포함됩니다.

> **Note**
>
> Amazon Inspector Code Security가 생성하는 보안 조사 결과는 이 통합에 사용할 수 없습니다. 그러나 Amazon Inspector 콘솔 및 Amazon Inspector API를 통해 이러한 특정 조사 결과에 액세스할 수 있습니다.

주제

- 에서 Amazon Inspector 조사 결과 보기 AWS Security Hub CSPM
- Security Hub CSPM과의 Amazon Inspector 통합 활성화 및 구성
- 조직 정책을 사용하여 Security Hub CSPM에서 Amazon Inspector 활성화
- 통합에서 조사 결과의 흐름 비활성화
- Security Hub CSPM에서 Amazon Inspector에 대한 보안 제어 보기

에서 Amazon Inspector 조사 결과 보기 AWS Security Hub CSPM

Security Hub CSPM에서 Amazon Inspector Classic 및 Amazon Inspector 조사 결과를 볼 수 있습니다.

> **Note**
>
> Amazon Inspector 조사 결과만 필터링하려면 필터 표시줄에 `"aws/inspector/ProductVersion": "2"`를 추가합니다. 이 필터는 Security Hub CSPM 대시보드에서 Amazon Inspector Classic 조사 결과를 제외합니다.

Amazon Inspector 조사 결과 예제


```
{
  "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"
}
```


Security Hub CSPM과의 Amazon Inspector 통합 활성화 및 구성

Security Hub CSPM을 활성화 AWS Security Hub CSPM 하여 와의 Amazon Inspector 통합을 활성화할 수 있습니다. https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-settingup.html Security Hub CSPM을 활성화하면 와의 Amazon Inspector 통합 AWS Security Hub CSPM 이 자동으로 활성화되고 Amazon Inspector는 Security AWS Finding Format(ASFF)을 사용하여 모든 조사 결과를 Security Hub CSPM으로 보내기 시작합니다.

조직 정책을 사용하여 Security Hub CSPM에서 Amazon Inspector 활성화

Security Hub CSPM 콘솔에서 직접 AWS Organizations 정책을 사용하여 조직 전체에서 Amazon Inspector 활성화를 관리할 수 있습니다. 이 중앙 집중식 접근 방식을 사용하면 조직 수준의 정책 관리를 통해 여러 계정에 대해 Amazon Inspector 스캔을 동시에 활성화할 수 있습니다.

조직 정책을 사용하여 Security Hub CSPM을 통해 Amazon Inspector 활성화를 관리하는 방법에 대한 자세한 지침은 AWS Security Hub CSPM 사용 설명서의 Security Hub CSPM에 대한 위임된 관리자 계정 관리를 참조하세요.

통합에서 조사 결과의 흐름 비활성화

Amazon Inspector가 Security Hub CSPM으로 조사 결과를 전송하지 못하도록 하려면 Security Hub CSPM 콘솔 또는 API 및 AWS CLI를 사용할 수 있습니다.

Security Hub CSPM에서 Amazon Inspector에 대한 보안 제어 보기

Security Hub CSPM은 지원되는 제품 AWS 및 타사 제품의 결과를 분석하고 규칙에 대해 자동화된 지속적 보안 검사를 실행하여 자체 결과를 생성합니다. 규칙은 보안 제어로 표시되며, 표준의 요구 사항이 충족되고 있는지 여부를 판단하는 데 도움이 됩니다.

Amazon Inspector는 보안 제어를 사용하여 Amazon Inspector 특성이 활성화되어 있는지 또는 활성화해야 하는지 여부를 확인합니다. 이러한 특성은 다음과 같습니다.

- Amazon EC2 스캔
- Amazon ECR 스캔
- Lambda 표준 스캔
- Lambda 코드 스캔

자세한 내용은 AWS Security Hub CSPM 사용 설명서에서 Amazon Inspector 제어를 참조하세요.

Amazon Inspector에서 지원되는 운영 체제 및 프로그래밍 언어

Amazon Inspector는 다음에 설치된 소프트웨어 애플리케이션을 스캔할 수 있습니다.

- Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스

> **Note**
>
> Amazon EC2 인스턴스의 경우 Amazon Inspector는 에이전트 기반 스캔을 지원하는 운영 체제의 패키지 취약성을 스캔할 수 있습니다. Amazon Inspector는 하이브리드 스캔을 지원하는 운영 체제 및 프로그래밍 언어의 패키지 취약성을 스캔할 수도 있습니다. Amazon Inspector는 도구 체인 취약성을 스캔하지 않습니다. 애플리케이션을 빌드하는 데 사용되는 프로그래밍 언어 컴파일러 버전에는 이러한 취약성이 발생합니다.

- Amazon Elastic Container Registry(Amazon ECR) 리포지토리에 저장된 컨테이너 이미지

> **Note**
>
> ECR 컨테이너 이미지의 경우, Amazon Inspector에서 운영 체제 및 프로그래밍 언어 패키지 취약성을 스캔할 수 있습니다. Amazon Inspector는 Chainguard 및 Minimus에서 제공하는 강화된 이미지도 지원합니다. Amazon Inspector는 애플리케이션을 빌드하는 데 사용되는 프로그래밍 언어 컴파일러 Rust버전인에서 도구 체인 취약성을 스캔하지 않습니다.

- AWS Lambda 함수

> **Note**
>
> Lambda 함수의 경우 Amazon Inspector에서 프로그래밍 언어 패키지 취약성 및 코드 취약성을 스캔할 수 있습니다. Amazon Inspector는 도구 체인 취약성을 스캔하지 않습니다. 애플리케이션을 빌드하는 데 사용되는 프로그래밍 언어 컴파일러 버전에는 이러한 취약성이 발생합니다.

Amazon Inspector는 리소스를 스캔할 때 50개 이상의 데이터 피드를 소싱하여 일반적인 취약성 및 노출(CVE)에 대한 조사 결과를 생성합니다. 이러한 소스의 예로는 공급업체 보안 권고, 데이터 피드

및 위협 인텔리전스 피드와 NVD(National Vulnerability Database) 및 MITRE가 있습니다. Amazon Inspector는 소스 피드의 취약성 데이터를 하루에 한 번 이상 업데이트합니다.

Amazon Inspector에서 리소스를 스캔하려면 리소스가 지원되는 운영 체제를 실행 중이거나 지원되는 프로그래밍 언어를 사용해야 합니다. 이 단원의 주제에는 Amazon Inspector에서 다양한 리소스 및 스캔 유형에 대해 지원하는 운영 체제, 프로그래밍 언어 및 런타임이 나열되어 있습니다. 또한 중단된 운영 체제도 나열되어 있습니다.

> **Note**
>
> 공급업체가 지원을 중단한 운영 체제에 대해서는 Amazon Inspector에서 제한된 지원만 제공할 수 있습니다.

주제

- 지원되는 운영 체제
- 중단된 운영 체제
- 지원되는 프로그래밍 언어
- 지원되는 런타임

지원되는 운영 체제

이 단원에는 Amazon Inspector에서 지원하는 운영 체제가 나열되어 있습니다.

지원되는 운영 체제: Amazon EC2 스캔

다음 표에는 Amazon EC2 인스턴스 스캔을 위해 Amazon Inspector에서 지원하는 운영 체제가 나열되어 있습니다. 운영 체제별 공급업체 보안 권고와 에이전트 기반 스캔 및 에이전트 없는 스캔을 지원하는 운영 체제가 지정되어 있습니다.

에이전트 기반 스캔 방법을 사용하는 경우 모든 적격 인스턴스에 대해 연속 스캔을 수행하도록 SSM 에이전트를 구성합니다. Amazon Inspector에서는 SSM 에이전트의 버전을 3.2.2086.0 이상으로 구성할 것을 권장합니다. 자세한 내용은 Amazon EC2 Systems Manager 사용 설명서에서 SSM 에이전트 작업을 참조하세요.

Linux 운영 체제 탐지는 기본 패키지 관리자 리포지토리(rpm 및 dpkg)에 대해서만 지원되며 타사 애플리케이션, 확장 지원 리포지토리(RHEL EUS, E4S, AUS 및 TUS) 및 선택적 리포지토리(애플리케이션

스트림)는 포함되지 않습니다. Amazon Inspector는 실행 중인 커널에서 취약성을 검사합니다. Ubuntu와 같은 일부 운영 체제의 경우 업그레이드가 활성 조사 결과에 표시되려면 재부팅이 필요합니다.

운영 체제	버전	공급업체 보안 권고	에이전트 없는 스캔 지원	에이전트 기반 스캔 지원
AlmaLinux	8	Errata CVE	예	예
AlmaLinux	9	Errata CVE	예	예
AlmaLinux	10	Errata CVE	아니요	예
Amazon Linux(AL2)	AL2	ALAS Errata CVE	예	예
Amazon Linux 2023(AL2023)	AL2023	ALAS Errata CVE	예	예
Bottlerocket	1.7.0 이상	Errata CVE	아니요	예
Debian Server(Bullseye)	11	DSA CVE	예	예
Debian Server(Bookworm)	12	DSA CVE	예	예
Debian Server(Trixie)	13	DSA CVE	예	예
Fedora	42	Errata CVE	예	예
OpenSUSE Leap	15.6	Errata CVE	예	예
Oracle Linux(Oracle)	8	Errata CVE	예	예
Oracle Linux(Oracle)	9	Errata CVE	예	예

운영 체제	버전	공급업체 보안 권고	에이전트 없는 스캔 지원	에이전트 기반 스캔 지원
Oracle Linux(Oracle)	10	Errata CVE	아니요	예
Red Hat Enterprise Linux(RHEL)	8	RHEL VEX CVE	예	예
Red Hat Enterprise Linux(RHEL)	9	RHEL VEX CVE	예	예
Red Hat Enterprise Linux(RHEL)	10	RHEL VEX CVE	아니요	예
Rocky Linux	8	Errata CVE	예	예
Rocky Linux	9	Errata CVE	예	예
Rocky Linux	10	Errata CVE	아니요	예
SUSE Linux Enterprise Server(SLES)	15.7	SUSE CVE	예	예
Ubuntu(Xenial)	16.04	USN, Ubuntu Pro(esm-infra 및 esm-apps)	예	예
Ubuntu(Bionic)	18.04	USN, Ubuntu Pro(esm-infra 및 esm-apps)	예	예
Ubuntu(Focal)	20.04	USN, Ubuntu Pro(esm-infra 및 esm-apps)	예	예

운영 체제	버전	공급업체 보안 권고	에이전트 없는 스캔 지원	에이전트 기반 스캔 지원
Ubuntu(Jammy)	22.04	USN, Ubuntu Pro(esm-infra 및 esm-apps)	예	예
Ubuntu (Noble Numbat)	24.04	USN, Ubuntu Pro (esm-infra & esm-apps)	예	예
Windows Server	2016	MSKB	아니요	예
Windows Server	2019	MSKB	아니요	예
Windows Server	2022	MSKB	아니요	예
Windows Server	2025	MSKB	아니요	예
macOS(Mojave)	10.14	APPLE-SA	아니요	예
macOS(Catalina)	10.15	APPLE-SA	아니요	예
macOS(Big Sur)	11	APPLE-SA	아니요	예
macOS(Monterey)	12	APPLE-SA	아니요	예
macOS(Ventura)	13	APPLE-SA	아니요	예
macOS(Sonoma)	14	APPLE-SA	아니요	예
macOS(Sequoia)	15	APPLE-SA	아니요	예

지원되는 운영 체제: Amazon Inspector를 사용한 Amazon ECR 스캔

다음 표에는 Amazon ECR 리포지토리에서 컨테이너 이미지 스캔을 위해 Amazon Inspector에서 지원하는 운영 체제가 나열되어 있습니다. 또한 운영 체제별 공급업체 보안 권고도 지정되어 있습니다.

운영 체제	버전	공급업체 보안 권고
AlmaLinux	8	Errata CVE
AlmaLinux	9	Errata CVE
AlmaLinux	10	Errata CVE
Alpine Linux (Alpine)	3.20	Errata CVE
Alpine Linux (Alpine)	3.21	Errata CVE
Alpine Linux (Alpine)	3.22	Errata CVE
Alpine Linux (Alpine)	3.23	Errata CVE
Amazon Linux (AL2)	AL2	CVE
Amazon Linux 2023 (AL2023)	AL2023	CVE
BusyBox	–	MITRE CVE
Chainguard	–	Errata CVE
Debian Server (Bullseye)	11	DSA CVE
Debian Server (Bookworm)	12	DSA CVE
Debian Server (Trixie)	13	DSA CVE
Echo	2	Errata CVE
Fedora	42	Errata CVE
Minimus	–	Errata CVE
OpenSUSE Leap	15.6	Errata CVE
Oracle Linux (Oracle)	8	Errata CVE
Oracle Linux (Oracle)	9	Errata CVE

운영 체제	버전	공급업체 보안 권고
Oracle Linux (Oracle)	10	Errata CVE
Photon OS	4	Errata CVE
Photon OS	5	Errata CVE
Red Hat Enterprise Linux (RHEL)	8	RHEL VEX CVE
Red Hat Enterprise Linux (RHEL)	9	RHEL VEX CVE
Red Hat Enterprise Linux (RHEL)	10	RHEL VEX CVE
Rocky Linux	8	Errata CVE
Rocky Linux	9	Errata CVE
Rocky Linux	10	Errata CVE
SUSE Linux Enterprise Server (SLES)	15.7	SUSE CVE
Ubuntu (Xenial)	16.04	USN, Ubuntu Pro (esm-infra & esm-apps)
Ubuntu (Bionic)	18.04	USN, Ubuntu Pro (esm-infra & esm-apps)
Ubuntu (Focal)	20.04	USN, Ubuntu Pro (esm-infra & esm-apps)
Ubuntu (Jammy)	22.04	USN, Ubuntu Pro (esm-infra & esm-apps)
Ubuntu (Noble Numbat)	24.04	USN, Ubuntu Pro (esm-infra & esm-apps)

운영 체제	버전	공급업체 보안 권고
Wolfi	–	Errata CVE

지원되는 운영 체제: CIS 스캔

다음 표에는 CIS 스캔을 위해 Amazon Inspector에서 지원하는 운영 체제가 나열되어 있습니다. 또한 운영 체제별 CIS 벤치마크 버전도 지정되어 있습니다.

> **Note**
>
> CIS 표준은 x86_64 운영 체제용입니다. 일부 검사는 평가되지 않거나 ARM 기반 리소스에 대한 잘못된 수정 지침을 반환할 수 있습니다.

운영 체제	버전	CIS 벤치마크 버전
Amazon Linux 2	AL2	3.0.0
Amazon Linux 2023	AL2023	1.0.0
Red Hat Enterprise Linux(RHEL)	8	3.0.0
Red Hat Enterprise Linux(RHEL)	9	2.0.0
Rocky Linux	8	2.0.0
Rocky Linux	9	1.0.0
SUSE Linux Enterprise Server	15	2.0.1
Ubuntu(Bionic)	18.04	2.2.0
Ubuntu(Focal)	20.04	3.0.0
Ubuntu(Jammy)	22.04	2.0.0

운영 체제	버전	CIS 벤치마크 버전
Ubuntu(Noble Numbat)	24.04	1.0.0
Windows Server	2016	3.0.0
Windows Server	2019	4.0.0
Windows Server	2022	4.0.0
Windows Server	2025	1.0.0

지원되는 운영 체제: Amazon Inspector Scan API

다음 표에는 Amazon Inspector Scan API에서 지원되는 운영 체제가 나열되어 있습니다. 자세한 내용은 Amazon Inspector V2 API 참조의 ScanSbom을 참조하세요.

운영 체제	버전
AlmaLinux 8	8
AlmaLinux	9
AlmaLinux	10
Alpine Linux	3.20
Alpine Linux	3.21
Alpine Linux	3.22
Alpine Linux	3.23
Amazon Linux	2
Amazon Linux	2023
Bottlerocket	–
BusyBox	1.36.0 이상

운영 체제	버전
Chainguard	–
Debian	11
Debian	12
Debian	13
Debian Sid	–
Echo	2
Fedora	42
Fedora	43
macOS	11+
MinimOS	–
OpenSUSE	15.6
Oracle Linux	8
Oracle Linux	9
Oracle Linux	10
Photon OS	4
Photon OS	5
Red Hat Enterprise Linux	8
Red Hat Enterprise Linux	9
Red Hat Enterprise Linux	10
Rocky Linux	8

운영 체제	버전
Rocky Linux	9
Rocky Linux	10
SUSE Server	15.7
Ubuntu	16.04
Ubuntu	18.04
Ubuntu	20.04
Ubuntu	22.04
Ubuntu	24.04
Ubuntu	25.10
Wolfi Linux	–

중단된 운영 체제

다음 표에는 중단된 운영 체제와 중단된 시간이 나와 있습니다.

Amazon Inspector가 중단된 운영 체제에 대한 전체 지원을 제공하지는 않지만 Amazon Inspector는 해당 운영 체제를 실행하는 Amazon EC2 인스턴스 및 Amazon ECR 컨테이너 이미지를 계속 스캔합니다. 보안 모범 사례로 지원되는 버전으로 이동하는 것이 좋습니다. Amazon Inspector가 중단된 운영 체제에 대해 생성하는 결과는 정보 제공 목적으로만 사용해야 합니다.

공급업체 정책에 따라 중단된 운영 체제는 더 이상 패치 업데이트를 받지 않습니다. 중단된 운영 체제에 대해서는 새로운 보안 권고가 릴리스되지 않을 수 있습니다. 공급업체는 표준 지원이 종료되는 운영 체제에 대해 기존 보안 권고 및 탐지를 피드에서 제거할 수 있습니다. 따라서 Amazon Inspector에서 알려진 CVE에 대한 조사 결과 생성이 중단될 수 있습니다.

운영 체제	버전	중단됨
Alpine Linux(Alpine)	3.2	2017년 5월 1일

운영 체제	버전	중단됨
Alpine Linux(Alpine)	3.3	2017년 11월 1일
Alpine Linux(Alpine)	3.4	2018년 5월 1일
Alpine Linux(Alpine)	3.5	2018년 11월 1일
Alpine Linux(Alpine)	3.6	2019년 5월 1일
Alpine Linux(Alpine)	3.7	2019년 11월 1일
Alpine Linux(Alpine)	3.8	2020년 5월 1일
Alpine Linux(Alpine)	3.9	2020년 11월 1일
Alpine Linux(Alpine)	3.10	2021년 5월 1일
Alpine Linux(Alpine)	3.11	2021년 11월 1일
Alpine Linux(Alpine)	3.12	2022년 5월 1일
Alpine Linux(Alpine)	3.13	2022년 11월 1일
Alpine Linux(Alpine)	3.14	2023년 5월 1일
Alpine Linux(Alpine)	3.15	2023년 11월 1일
Alpine Linux(Alpine)	3.16	2024년 5월 23일
Alpine Linux(Alpine)	3.17	2024년 11월 22일
Alpine Linux(Alpine)	3.18	2025년 5월 9일
Alpine Linux(Alpine)	3.19	2025년 11월 1일
Amazon Linux(AL1)	2012	2021년 12월 31일
CentOS Linux(CentOS)	7	2024년 6월 30일
CentOS Linux(CentOS)	8	2021년 12월 31일

운영 체제	버전	중단됨
Debian Server(Jessie)	8	2020년 6월 30일
Debian Server(Stretch)	9	2022년 6월 30일
Debian Server(Buster)	10	2024년 6월 30일
Fedora	33	2021년 11월 30일
Fedora	34	2022년 6월 7일
Fedora	35	2022년 12월 13일
Fedora	36	2023년 5월 16일
Fedora	37	2023년 12월 15일
Fedora	38	2024년 5월 21일
Fedora	39	2024년 11월 26일
Fedora	40	2025년 5월 13일
Fedora	41	2025년 11월 19일
OpenSUSE Leap	15.2	2021년 12월 1일
OpenSUSE Leap	15.3	2022년 12월 1일
OpenSUSE Leap	15.4	2023년 12월 7일
OpenSUSE Leap	15.5	2024년 12월 31일
Oracle Linux(Oracle)	6	2021년 3월 1일
Oracle Linux(Oracle)	7	2024년 12월 31일
Photon OS	2	2021년 12월 2일
Photon OS	3	2024년 3월 1일

운영 체제	버전	중단됨
Red Hat Enterprise Linux(RHEL)	6	2020년 6월 30일
Red Hat Enterprise Linux(RHEL)	7	2024년 6월 30일
SUSE Linux Enterprise Server(SLES)	12	2016년 6월 30일
SUSE Linux Enterprise Server(SLES)	12.1	2017년 5월 31일
SUSE Linux Enterprise Server(SLES)	12.2	2018년 3월 31일
SUSE Linux Enterprise Server(SLES)	12.3	2019년 6월 30일
SUSE Linux Enterprise Server(SLES)	12.4	2020년 6월 30일
SUSE Linux Enterprise Server(SLES)	12.5	2024년 10월 31일
SUSE Linux Enterprise Server(SLES)	15	2019년 12월 31일
SUSE Linux Enterprise Server(SLES)	15.1	2021년 1월 31일
SUSE Linux Enterprise Server(SLES)	15.2	2021년 12월 31일
SUSE Linux Enterprise Server(SLES)	15.3	2022년 12월 31일
SUSE Linux Enterprise Server(SLES)	15.4	2023년 12월 31일
SUSE Linux Enterprise Server(SLES)	15.5	2024년 12월 31일
SUSE Linux Enterprise Server(SLES)	15.6	2025년 12월 31일
Ubuntu(Trusty)	12.04	2017년 4월 28일
Ubuntu(Trusty)	14.04	2024년 4월 1일
Ubuntu(Groovy)	20.10	2021년 7월 22일
Ubuntu(Hirsute)	21.04	2022년 1월 20일
Ubuntu(Impish)	21.10	2022년 7월 31일

운영 체제	버전	중단됨
Ubuntu(키네틱)	22.10	2023년 7월 20일
Ubuntu(Lunar Lobster)	23.04	2024년 1월 25일
Ubuntu(Mantic Minotaur)	23.10	2024년 7월 11일
Ubuntu(Oracular Oriole)	24.10	2025년 7월 10일
Ubuntu(Plucky Puffin)	25.04	2026년 1월 15일
Windows Server	2012	2023년 10월 10일
Windows Server	2012 R2	2023년 10월 10일

지원되는 프로그래밍 언어

이 섹션에는 Amazon Inspector에서 지원하는 프로그래밍 언어가 나열되어 있습니다.

지원되는 프로그래밍 언어: Amazon EC2 에이전트 없는 스캔

Amazon Inspector는 현재 적격 Amazon EC2 인스턴스에 대해 에이전트 없는 스캔을 수행할 때 다음 프로그래밍 언어를 지원합니다. 자세한 내용은 에이전트 없는 스캔을 참조하세요.

> **Note**
>
> Amazon Inspector는 Go 및 Rust에서 도구 체인 취약성을 스캔하지 않습니다. 애플리케이션을 빌드하는 데 사용되는 프로그래밍 언어 컴파일러 버전에는 이러한 취약성이 발생합니다.

- C#
- Go
- Java
- JavaScript
- PHP
- Python

- Ruby
- Rust

지원되는 프로그래밍 언어: Amazon EC2 심층 검사

Amazon Inspector는 현재 Amazon EC2 Linux 인스턴스에 대해 심층 검사 스캔을 수행할 때 다음 프로그래밍 언어를 지원합니다. 자세한 내용은 Linux 기반 Amazon EC2 인스턴스에 대한 Amazon Inspector 심층 검사를 참조하세요.

- Java(.ear, .jar, .par, .war 아카이브 형식)
- JavaScript
- Python

Amazon Inspector는 Systems Manager Distributor를 사용하여 Amazon EC2 인스턴스의 심층 검사용 플러그인을 배포합니다.

> **Note**
>
> Bottlerocket 운영 체제에서는 심층 검사가 지원되지 않습니다.

심층 검사 스캔을 수행하려면 Systems Manager Distributor 및 Amazon Inspector에서 Amazon EC2 인스턴스 운영 체제를 지원해야 합니다. Systems Manager Distributor에서 지원하는 운영 체제에 대한 자세한 내용은 Systems Manager 사용 설명서에서 지원되는 패키지 플랫폼 및 아키텍처를 참조하세요.

지원되는 프로그래밍 언어: Amazon ECR 스캔

Amazon Inspector는 현재 Amazon ECR 리포지토리에서 컨테이너 이미지를 스캔할 때 다음 프로그래밍 언어를 지원합니다.

> **Note**
>
> Amazon Inspector는 Rust에서 도구 체인 취약성을 스캔하지 않습니다. 애플리케이션을 빌드하는 데 사용되는 프로그래밍 언어 컴파일러 버전에는 이러한 취약성이 발생합니다. Chainguard 라이브러리를 사용하는 Python 애플리케이션의 경우 Amazon Inspector는 백포트 보안 수정 사항을 인식하고 조사 결과에서 제외합니다.

- C#
- Go
- Go 도구 체인
- Java
- Java JDK
- JavaScript
- PHP
- Python (Chainguard라이브러리 포함)
- Ruby
- Rust

지원되는 런타임

이 단원에는 Amazon Inspector에서 지원하는 런타임이 나열되어 있습니다.

지원되는 런타임: Amazon Inspector Lambda 표준 스캔

Amazon Inspector Lambda 표준 스캔은 현재 타사 소프트웨어 패키지에 취약성이 있는지 Lambda 함수를 스캔할 때 사용할 수 있는 프로그래밍 언어에 대해 다음과 같은 런타임을 지원합니다.

> **Note**
>
> Amazon Inspector는 Rust에서 도구 체인 취약성을 스캔하지 않습니다. 애플리케이션을 빌드하는 데 사용되는 프로그래밍 언어 컴파일러 버전에는 이러한 취약성이 발생합니다.

- Go
  - go1.x
- Java
  - java8
  - java8.al2
  - java11
  - java17

  - java21
- .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
- Ruby
  - ruby2.7
  - ruby3.2
  - ruby3.3
- Custom runtimes
  - AL2
  - AL2023

지원되는 런타임: Amazon Inspector Lambda 코드 스캔

Amazon Inspector Lambda 코드 스캔은 현재 코드에 취약성이 있는지 Lambda 함수를 스캔할 때 사용할 수 있는 프로그래밍 언어에 대해 다음과 같은 런타임을 지원합니다.

- Java
  - java8
  - java8.al2
  - java11
  - java17
- .NET
  - .NET 6
  - .NET 8
- Node.js
  - nodejs12.x
  - nodejs14.x
  - nodejs16.x
  - nodejs18.x
  - nodejs20.x
- Python
  - python3.7
  - python3.8
  - python3.9
  - python3.10
  - python3.11
  - python3.12
- Ruby
  - ruby2.7
  - ruby3.2
  - ruby3.3

Amazon Inspector 비활성화

Amazon Inspector 콘솔 또는 Amazon Inspector API를 통해 Amazon Inspector를 비활성화할 수 있습니다. 계정에 대한 모든 스캔 유형을 비활성화하면 해당 계정에 대해 Amazon Inspector가 자동으로 비활성화됩니다.

계정에 대해 Amazon Inspector를 비활성화하면 해당 계정에 대한 모든 스캔 유형이 비활성화됩니다. 또한 계정에 대한 필터, 억제 규칙 및 조사 결과를 포함한 모든 Amazon Inspector 스캔 설정이 삭제됩니다.

Amazon Inspector Amazon EC2 스캔을 비활성화하면 Amazon Inspector에서 다음 SSM 연결이 삭제됩니다.

- `InspectorDistributor-do-not-delete`
- `InspectorInventoryCollection-do-not-delete`
- `InvokeInspectorSsmPlugin-do-not-delete`. 또한 이 연결을 통해 설치된 Amazon Inspector SSM 플러그인도 모든 Windows 호스트에서 제거됩니다. 자세한 내용은 Windows EC2 인스턴스 스캔 단원을 참조하십시오.

> **Note**
>
> Amazon Inspector 를 비활성화하면 더 이상 서비스 요금이 발생하지 않습니다. 하지만 언제든지 Amazon Inspector를 다시 활성화할 수 있습니다.

다양한 리소스에 대한 스캔 유형을 비활성화하는 자세한 방법은 스캔 유형 비활성화를 참조하세요.

사전 조건

계정 유형에 따라 다음 사항을 고려하세요.

- 독립 실행형 Amazon Inspector 계정인 경우 언제든지 Amazon Inspector를 비활성화할 수 있습니다.
- 다중 계정 환경의 멤버 계정인 경우 Amazon Inspector를 비활성화할 수 없습니다. Amazon Inspector를 비활성화하려면 조직의 위임된 관리자에게 문의해야 합니다.
- 조직의 위임된 관리자인 경우 Amazon Inspector를 비활성화하기 전에 모든 멤버 계정의 연결을 해제해야 합니다.

- 계정의 Amazon Inspector 활성화가 AWS Organizations 정책에 의해 관리되는 경우 Amazon Inspector 콘솔 또는 API를 통해 정책 관리형 스캔 유형을 비활성화할 수 없습니다. Amazon Inspector 스캔 유형을 비활성화하려면 AWS Organizations 콘솔 또는 API를 통해 명시적으로 비활성화하도록 조직 정책을 수정해야 합니다. Amazon Inspector 콘솔 또는 API를 통해 조직 정책에서 관리하지 않는 스캔 유형을 비활성화할 수 있습니다.

> **Note**
>
> 위임된 관리자가 Amazon Inspector를 비활성화하면 조직에 대한 자동 활성화 기능이 비활성화됩니다.

조직 정책에서 관리하는 Amazon Inspector 비활성화

AWS Organizations 정책을 통해 계정에서 Amazon Inspector가 활성화된 경우 AWS Organizations 콘솔 또는 API를 사용하여 Inspector를 비활성화해야 합니다. 멤버 계정과 위임된 관리자는 Amazon Inspector 콘솔 또는 API를 통해 정책 관리형 스캔 유형을 비활성화할 수 없습니다.

정책 관리형 계정에 대해 Amazon Inspector를 비활성화하려면:

정책 관리형 Amazon Inspector 활성화를 비활성화하려면

1. AWS Organizations 관리 계정 또는 정책 관리자 계정에 로그인합니다.
2. 조직 정책을 수정하여 Inspector를 비활성화하려는 리전에서 스캔 유형을 비활성화로 명시적으로 설정합니다. 비활성화하려는 스캔 유형에 대해 비활성화된 리전을 지정하려면 정책 콘텐츠를 업데이트해야 합니다.
3. AWS Organizations 는 정책 변경 사항을 자동으로 적용하고 Amazon Inspector는 영향을 받는 계정에서 지정된 스캔 유형을 비활성화합니다.

조직 정책 수정 또는 분리에 대한 자세한 지침은 Amazon Inspector 정책 AWS Organizations 설명서를 참조하세요.

> **Note**
>
> 계정에서 조직 정책을 분리하면 해당 계정은 현재 Amazon Inspector 설정(마지막으로 적용된 정책에 따라 활성화 또는 비활성화됨)을 유지합니다. 계정은 더 이상 정책에 의해 관리되지 않으며 Amazon Inspector 설정을 독립적으로 또는 위임된 관리자를 통해 관리할 수 있습니다.

Amazon Inspector 비활성화

> **Note**
>
> Amazon Inspector 를 비활성화하기 전에 조사 결과 내보내기를 고려하세요.

Console

Amazon Inspector를 비활성화하려면

1. 자격 증명을 사용하여 로그인한 다음 Amazon Inspector 콘솔(https://console.aws.amazon.com/inspector/v2/home)을 엽니다.
2. 페이지 오른쪽 상단의 AWS 리전 선택기를 사용하여 Amazon Inspector를 비활성화할 리전을 선택합니다.
3. 탐색 창에서 일반 설정을 선택합니다.
4. Inspector 비활성화를 선택합니다.
5. 확인 메시지가 표시되면 텍스트 상자에 deactivate를 입력한 다음 Inspector 비활성화를 선택합니다.
6. (권장) Amazon Inspector를 비활성화할 각 리전에 대해 이 단계를 반복합니다.

API

Disable API 작업을 실행합니다. 요청에 비활성화할 계정 ID를 제공하고 `EC2, ECR, LAMBDA`를 `resourceTypes`로 제공하여 모든 스캔을 비활성화하면 해당 계정이 비활성화됩니다.

Amazon Inspector 할당량

이 단원에는 AWS 리전별 Amazon Inspector 할당량이 나열되어 있습니다.

리소스	기본값	설명
멤버 계정	10,000개	Amazon Inspector 위임 관리자 계정과 연결된 최대 멤버 계정 수입니다. 이 한도는 AWS Organizations의 할당량을 기준으로 합니다.
억제 규칙	500	리전별로 AWS 계정당 저장된 최대 억제 규칙 수입니다. 할당량 증가를 요청할 수 없습니다.
Amazon EC2 네트워크 조사 결과	10,000개	AWS 계정당 최대 Amazon EC2 네트워크 조사 결과 수입니다. 할당량 증가를 요청할 수 없습니다.
CIS 스캔 구성	500	CIS 스캔 구성의 최대 개수입니다. 할당량 증가를 요청할 수 없습니다.

Amazon Inspector Classic과 관련된 할당량 목록은 AWS 일반 참조에서 Amazon Inspector Classic 서비스 할당량을 참조하세요. AWS Organizations와 관련된 할당량 목록은 AWS 일반 참조에서 AWS Organizations 서비스 할당량을 참조하세요.

리전 및 엔드포인트

이 주제에는 Amazon Inspector와 Amazon Inspector 스캔에 대한 엔드포인트를 보여주는 표가 포함되어 있습니다. 또한 Amazon Inspector 기능을 AWS 리전 지원하는 테이블도 포함되어 있습니다. Amazon Inspector를 사용할 수 AWS 리전 있는를 보려면의 Amazon Inspector 엔드포인트 및 할당량을 참조하세요Amazon Web Services 일반 참조.

Amazon Inspector의 서비스 엔드포인트

다음 표에는 Amazon Inspector의 서비스 엔드포인트가 나와 있습니다. Amazon Inspector 스캔 엔드포인트의 명명 규칙은 `inspector2.Region.amazonaws.com`입니다.

리전 이름	리전	엔드포인트	프로토콜
미국 동부(오하이오)	us-east-2	inspector2.us-east-2.amazonaws.com	HTTPS
		inspector2-fips.us-east-2.amazonaws.com	HTTPS
미국 동부(버지니아 북부)	us-east-1	inspector2.us-east-1.amazonaws.com	HTTPS
		inspector2-fips.us-east-1.amazonaws.com	HTTPS
미국 서부(캘리포니아 북부)	us-west-1	inspector2.us-west-1.amazonaws.com	HTTPS
		inspector2-fips.us-west-1.amazonaws.com	HTTPS
미국 서부(오레곤)	us-west-2	inspector2.us-west-2.amazonaws.com	HTTPS
		inspector2-fips.us-west-2.amazonaws.com	HTTPS
아프리카(케이프타운)	af-south-1	inspector2.af-south-1.amazonaws.com	HTTPS
아시아 태평양(홍콩)	ap-east-1	inspector2.ap-east-1.amazonaws.com	HTTPS

리전 이름	리전	엔드포인트	프로토콜	
아시아 태평양(하이데라바드)	ap-south-2	inspector2.ap-south-2.amazonaws.com	HTTPS	
아시아 태평양(자카르타)	ap-southeast-3	inspector2.ap-southeast-3.amazonaws.com	HTTPS	
아시아 태평양(말레이시아)	ap-southeast-5	inspector2.ap-southeast-5.amazonaws.com	HTTPS	
아시아 태평양(멜버른)	ap-southeast-4	inspector2.ap-southeast-4.amazonaws.com	HTTPS	
아시아 태평양(뭄바이)	ap-south-1	inspector2.ap-south-1.amazonaws.com	HTTPS	
아시아 태평양(오사카)	ap-northeast-3	inspector2.ap-northeast-3.amazonaws.com	HTTPS	
아시아 태평양(서울)	ap-northeast-2	inspector2.ap-northeast-2.amazonaws.com	HTTPS	
아시아 태평양(싱가포르)	ap-southeast-1	inspector2.ap-southeast-1.amazonaws.com	HTTPS	
아시아 태평양(시드니)	ap-southeast-2	inspector2.ap-southeast-2.amazonaws.com	HTTPS	

리전 이름	리전	엔드포인트	프로토콜	
아시아 태평양(태국)	ap-southeast-7	inspector2.ap-southeast-7.amazonaws.com	HTTPS	
아시아 태평양(도쿄)	ap-northeast-1	inspector2.ap-northeast-1.amazonaws.com	HTTPS	
캐나다(중부)	ca-central-1	inspector2.ca-central-1.amazonaws.com	HTTPS	
캐나다 서부(캘거리)	ca-west-1	inspector2.ca-west-1.amazonaws.com	HTTPS	
유럽(프랑크푸르트)	eu-central-1	inspector2.eu-central-1.amazonaws.com	HTTPS	
유럽(아일랜드)	eu-west-1	inspector2.eu-west-1.amazonaws.com	HTTPS	
유럽(런던)	eu-west-2	inspector2.eu-west-2.amazonaws.com	HTTPS	
유럽(밀라노)	eu-south-1	inspector2.eu-south-1.amazonaws.com	HTTPS	
유럽(파리)	eu-west-3	inspector2.eu-west-3.amazonaws.com	HTTPS	
유럽(스페인)	eu-south-2	inspector2.eu-south-2.amazonaws.com	HTTPS	
유럽(스톡홀름)	eu-north-1	inspector2.eu-north-1.amazonaws.com	HTTPS	

리전 이름	리전	엔드포인트	프로토콜
유럽(취리히)	eu-central-2	inspector2.eu-central-2.amazonaws.com	HTTPS
이스라엘(텔아비브)	il-central-1	inspector2.il-central-1.amazonaws.com	HTTPS
멕시코(중부)	mx-central-1	inspector2.mx-central-1.amazonaws.com	HTTPS
중동(바레인)	me-south-1	inspector2.me-south-1.amazonaws.com	HTTPS
중동(UAE)	me-central-1	inspector2.me-central-1.amazonaws.com	HTTPS
남아메리카(상파울루)	sa-east-1	inspector2.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud(미국 동부)	us-gov-east-1	inspector2.us-gov-east-1.amazonaws.com inspector2-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS
AWS GovCloud(미국 서부)	us-gov-west-1	inspector2.us-gov-west-1.amazonaws.com inspector2-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS

Amazon Inspector 스캔 API용 엔드포인트

다음 표에는 Amazon Inspector 스캔 API를 직접적으로 호출할 때 사용할 수 있는 리전 엔드포인트가 나와 있습니다. API를 사용할 때는 엔드포인트와 현재 인증된 리전에 해당하는 AWS 리전을 제공해야 합니다.

Amazon Inspector 스캔 엔드포인트의 명명 규칙은 `inspector-scan.region.amazonaws.com`입니다. 예를 들어, `us-west-2`에서 인증된 경우 엔드포인트 `inspector-scan.us-west-2.amazonaws.com`을 사용하여 `inspector-scan` API를 직접적으로 호출합니다.

리전 이름	리전	엔드포인트	프로토콜
미국 동부(오하이오)	us-east-2	inspector-scan.us-east-2.amazonaws.com	HTTPS
		inspector-scan-fips.us-east-2.amazonaws.com	HTTPS
미국 동부(버지니아 북부)	us-east-1	inspector-scan.us-east-1.amazonaws.com	HTTPS
		inspector-scan-fips.us-east-1.amazonaws.com	HTTPS
미국 서부(캘리포니아 북부)	us-west-1	inspector-scan.us-west-1.amazonaws.com	HTTPS
		inspector-scan-fips.us-west-1.amazonaws.com	HTTPS
미국 서부(오레곤)	us-west-2	inspector-scan.us-west-2.amazonaws.com	HTTPS
		inspector-scan-fips.us-west-2.amazonaws.com	HTTPS
Africa (Cape Town)	af-south-1	inspector-scan.af-south-1.amazonaws.com	HTTPS
아시아 태평양(홍콩)	ap-east-1	inspector-scan.ap-east-1.amazonaws.com	HTTPS
아시아 태평양(하이데라바드)	ap-south-2	inspector-scan.ap-south-2.amazonaws.com	HTTPS
아시아 태평양(자카르타)	ap-southeast-3	inspector-scan.ap-southeast-3.amazonaws.com	HTTPS

리전 이름	리전	엔드포인트	프로토콜	
아시아 태평양(말레이시아)	ap-southeast-5	inspector-scan.ap-southeast-5.amazonaws.com	HTTPS	
아시아 태평양(멜버른)	ap-southeast-4	inspector-scan.ap-southeast-4.amazonaws.com	HTTPS	
아시아 태평양(뭄바이)	ap-south-1	inspector-scan.ap-south-1.amazonaws.com	HTTPS	
아시아 태평양(오사카)	ap-northeast-3	inspector-scan.ap-northeast-3.amazonaws.com	HTTPS	
아시아 태평양(서울)	ap-northeast-2	inspector-scan.ap-northeast-2.amazonaws.com	HTTPS	
아시아 태평양(싱가포르)	ap-southeast-1	inspector-scan.ap-southeast-1.amazonaws.com	HTTPS	
아시아 태평양(시드니)	ap-southeast-2	inspector-scan.ap-southeast-2.amazonaws.com	HTTPS	
아시아 태평양(태국)	ap-southeast-7	inspector-scan.ap-southeast-7.amazonaws.com	HTTPS	
아시아 태평양(도쿄)	ap-northeast-1	inspector-scan.ap-northeast-1.amazonaws.com	HTTPS	

리전 이름	리전	엔드포인트	프로토콜
캐나다(중부)	ca-central-1	inspector-scan.ca-central-1.amazonaws.com	HTTPS
캐나다 서부(캘거리)	ca-west-1	inspector-scan.ca-west-1.amazonaws.com	HTTPS
유럽(프랑크푸르트)	eu-central-1	inspector-scan.eu-central-1.amazonaws.com	HTTPS
유럽(아일랜드)	eu-west-1	inspector-scan.eu-west-1.amazonaws.com	HTTPS
유럽(런던)	eu-west-2	inspector-scan.eu-west-2.amazonaws.com	HTTPS
유럽(밀라노)	eu-south-1	inspector-scan.eu-south-1.amazonaws.com	HTTPS
유럽(파리)	eu-west-3	inspector-scan.eu-west-3.amazonaws.com	HTTPS
유럽(스페인)	eu-south-2	inspector-scan.eu-south-2.amazonaws.com	HTTPS
유럽(스톡홀름)	eu-north-1	inspector-scan.eu-north-1.amazonaws.com	HTTPS
유럽(취리히)	eu-central-2	inspector-scan.eu-central-2.amazonaws.com	HTTPS
이스라엘(텔아비브)	il-central-1	inspector-scan.il-central-1.amazonaws.com	HTTPS
멕시코(중부)	mx-central-1	inspector-scan.mx-central-1.amazonaws.com	HTTPS

리전 이름	리전	엔드포인트	프로토콜
중동(바레인)	me-south-1	inspector-scan.me-south-1.amazonaws.com	HTTPS
중동(UAE)	me-central-1	inspector-scan.me-central-1.amazonaws.com	HTTPS
남아메리카(상파울루)	sa-east-1	inspector-scan.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud(미국 동부)	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(미국 서부)	us-gov-west-1	inspector-scan.us-gov-west-1.amazonaws.com inspector-scan-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS

리전별 특성 가용성

이 섹션에서는 AWS 리전별 Amazon Inspector 특성의 사용 가능 여부에 대해 설명합니다.

Amazon EC2 리전에 대한 에이전트 없는 EC2 스캔

다음 표에는 현재 Amazon EC2에 대한 에이전트 없는 스캔을 사용할 수 AWS 리전 있는가 나와 있습니다.

리전 이름	리전 코드
미국 동부(버지니아 북부)	us-east-1
미국 동부(오하이오)	us-east-2
미국 서부(캘리포니아 북부)	us-west-1

리전 이름	리전 코드
미국 서부(오리건)	us-west-2
아프리카(케이프타운)	af-south-1
아시아 태평양(홍콩)	ap-east-1
아시아 태평양(도쿄)	ap-northeast-1
아시아 태평양(서울)	ap-northeast-2
아시아 태평양(오사카)	ap-northeast-3
아시아 태평양(뭄바이)	ap-south-1
아시아 태평양(하이데라바드)	ap-south-2
아시아 태평양(싱가포르)	ap-southeast-1
아시아 태평양(시드니)	ap-southeast-2
아시아 태평양(자카르타)	ap-southeast-3
아시아 태평양(멜버른)	ap-southeast-4
아시아 태평양(말레이시아)	ap-southeast-5
아시아 태평양(태국)	ap-southeast-7
캐나다(중부)	ca-central-1
캐나다 서부(캘거리)	ca-west-1
유럽(스톡홀름)	eu-north-1
유럽(프랑크푸르트)	eu-central-1
유럽(취리히)	eu-central-2
유럽(아일랜드)	eu-west-1

리전 이름	리전 코드
유럽(런던)	eu-west-2
유럽(파리)	eu-west-3
유럽(밀라노)	eu-south-1
유럽(스페인)	eu-south-2
이스라엘(텔아비브)	il-central-1
중동(UAE)	me-central-1
중동(바레인)	me-south-1
멕시코(중부)	mx-central-1
남아메리카(상파울루)	sa-east-1
AWS GovCloud(미국 동부)	us-gov-east-1
AWS GovCloud(미국 서부)	us-gov-west-1

Lambda 코드 스캔 리전

다음 표에는 현재 Lambda 코드 스캔을 사용할 수 AWS 리전 있는가 나와 있습니다.

리전 이름	리전 코드
미국 동부(버지니아 북부)	us-east-1
미국 서부(오리건)	us-west-2
미국 동부(오하이오)	us-east-2
아시아 태평양(시드니)	ap-southeast-2
아시아 태평양(도쿄)	ap-northeast-1
유럽(프랑크푸르트)	eu-central-1

리전 이름	리전 코드
유럽(아일랜드)	eu-west-1
유럽(런던)	eu-west-2
유럽(스톡홀름)	eu-north-1
아시아 태평양(싱가포르)	ap-southeast-1

> **Important**
>
> Lambda 코드 스캔을 사용할 수 없는 AWS 리전 에서 Amazon Inspector Enable API를 사용하여 Lambda 코드 스캔을 활성화하려고 하면 다음과 같은 액세스 거부 오류가 발생합니다.
>
> ```
> An error occurred (AccessDeniedException) when calling the Enable operation:
>  Lambda code scanning is not supported in unsupported-AWS ##
> ```

Amazon Inspector 코드 보안 리전

다음 표에는 Amazon Inspector 코드 보안을 현재 사용할 수 AWS 리전 있는가 나와 있습니다.

리전 이름	리전 코드
미국 동부(버지니아 북부)	us-east-1
미국 서부(오리건)	us-west-2
미국 동부(오하이오)	us-east-2
아시아 태평양(시드니)	ap-southeast-2
아시아 태평양(도쿄)	ap-northeast-1
유럽(프랑크푸르트)	eu-central-1
유럽(아일랜드)	eu-west-1

리전 이름	리전 코드
유럽(런던)	eu-west-2
유럽(스톡홀름)	eu-north-1
아시아 태평양(싱가포르)	ap-southeast-1

AWS GovCloud (US) 리전

자세한 내용은 AWS GovCloud (US) 사용 설명서에서 Amazon Inspector 섹션을 참조하세요.

문서 기록

다음 표에서는 2021년 11월부터 적용되는 Amazon Inspector 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 설명서 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하면 됩니다.

Amazon Inspector 제품 업데이트

변경 사항	설명	날짜
Amazon Inspector SBOM 생성기 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기가 CVE-2026-25679, CVE-2026-27142 및 CVE-2026-27139에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 알고 있습니다. Amazon Inspector SBOM 생성기가 이러한 취약성의 영향을 받지 않는 것으로 확인되었습니다. 이 취약성은 Amazon Inspector SBOM 생성기 버전을 1.11.2 이상으로 업그레이드하여 해결할 수 있습니다.	2026년 3월 11일
Amazon Inspector SBOM 생성기 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기가에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 알고 있습니다CVE-2025-15558 . Amazon Inspector SBOM 생성기는 CVE-2025-15558의 영향을 받지 않는 것으로 확인되었습니다. 이 취약성은 Amazon Inspector SBOM 생성기 버전을 1.11.1 이상으로 업그레이드하여 해결할 수 있습니다.	2026년 3월 5일

Amazon Inspector SBOM 생성기 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기가에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 알고 있습니다`CVE-2025-68121` . Amazon Inspector SBOM 생성기는 CVE-2025-68121의 영향을 받지 않는 것으로 확인되었습니다. 이 취약성은 Amazon Inspector SBOM 생성기 버전을 1.11.0 이상으로 업그레이드하여 해결할 수 있습니다.	2026년 3월 2일
새 관리형 정책	Amazon Inspector는 Amazon Inspector 원격 측정 작업에 대한 권한을 `AmazonInspector2ManagedTelemetryPolicy` 부여하는 새로운 관리형 정책을 릴리스하여 서비스가 취약성 스캔을 위해 패키지 인벤토리 데이터를 수집하고 전송할 수 있도록 했습니다. 자세한 내용은 AWS 관리형 정책에 대한 Amazon Inspector 업데이트를 참조하세요.	2026년 2월 5일

업데이트된 정책	Amazon Inspector는 AmazonInspector2ServiceRolePolicy 라는 서비스 연결 역할에 새 권한을 추가합니다. Amazon Inspector가 네트워크 연결성 분석을 위한 방화벽 메타데이터 Amazon Inspector를 설명할 수 있는 새 권한을 추가했습니다. 또한 Amazon Inspector는 Amazon Inspector가 SSM 문서 와의 SSM 연결을 생성, 업데이트 및 시작할 수 있도록 추가 리소스 범위 조정을 추가했습니다AWS-ConfigureAWSPackage . 자세한 내용은 Amazon Inspector에 대한 서비스 연결 역할 권한을 참조하세요.	2026년 2월 3일
Amazon Inspector SSM 플러그인 및 Amazon Inspector SBOM 생성기 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인과 Amazon Inspector SBOM 생성기가에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 알고 있습니다CVE-2025-61728, CVE-2025-61730, and CVE-2025-61726 . 이러한 취약성은 Amazon Inspector SSM 플러그인 버전을 1.0.2327.0 또는 Amazon Inspector SBOM 생성기 1.10.1 이상으로 업그레이드하여 해결할 수 있습니다.	2026년 1월 29일

Amazon Inspector SSM 플러그인 및 Amazon Inspector SBOM 생성기 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인과 Amazon Inspector SBOM 생성기가에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 알고 있습니다CVE-2025-61729 . 이러한 애플리케이션은이 CVE의 영향을 받지 않는 것으로 확인되었습니다. 현재 이 탐지를 해결하기 위한 개선 작업을 진행하고 있습니다. 그동안 고객은이 취약성을 무시하거나 억제할 수 있습니다.	2025년 12월 3일
Amazon Inspector SBOM 생성기에 대한 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기가 CVE-2025-47914 및에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 알고 있습니다CVE-2025-58181 . Amazon Inspector SBOM 생성기는 이러한 CVEs의 영향을 받지 않는 것으로 확인되었습니다. 현재 이러한 탐지를 해결하기 위한 개선 작업을 진행하고 있습니다. 그 동안 고객은 이러한 취약성을 무시하거나 억제할 수 있습니다.	2025년 11월 20일

새로운 특성	Amazon Inspector는 이제 조직 계정 전반의 중앙 집중식 활성화 및 거버넌스에 대한 AWS Organizations 정책을 지원합니다. 조직 정책을 사용하면 조직 전체에서 Amazon Inspector 스캔 유형을 자동으로 활성화하고 무단 수정을 방지할 수 있습니다. 자세한 내용은 시작하기 자습서 및 여러 계정 관리를 참조하세요.	2025년 11월 19일
Amazon Inspector SBOM 생성기에 대한 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기가에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인지하고 있습니다CVE-2025-47913 . Amazon Inspector SBOM 생성기가이 CVE의 영향을 받지 않는 것으로 확인되었으며이 감지를 해결하기 위해 업데이트가 배포되었습니다.	2025년 11월 14일
업데이트된 정책	Amazon Inspector는 관리형 정책 AmazonInspector2FullAccess_v2 및에 새 권한을 추가합니다AmazonInspector2ReadOnlyAccess . 권한을 통해 Amazon Inspector 조직 정책 및 정책을 통해 설정된 위임 구성을 볼 수 AWS Organizations 있습니다. 자세한 내용은 Amazon Inspector를 위한AWS 관리형 정책을 참조하세요.	2025년 11월 14일

Amazon Inspector SBOM 생성기에 대한 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기 버전을 업데이트합니다. 자세한 내용은 Amazon Inspector SBOM 생성기의 이전 버전을 참조하세요.	2025년 11월 11일
업데이트된 정책	Amazon Inspector는 `AmazonInspector2ServiceRolePolicy` 라는 서비스 연결 역할에 새 권한을 추가합니다. 권한을 통해 Amazon Inspector AWS Organizations 정책은 Amazon Inspector의 활성화 및 비활성화를 적용할 수 있습니다. 자세한 내용은 Amazon Inspector에 대한 서비스 연결 역할 권한을 참조하세요.	2025년 11월 10일
Amazon Inspector SBOM 생성기에 대한 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기가 `CVE-2025-58188` 및 `CVE-2025-61725` 에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인식합니다. Amazon Inspector SBOM 생성기는 이러한 CVE의 영향을 받지 않는 것으로 확인되었으며 Amazon Inspector는 Amazon Inspector SBOM 생성기 버전을 업데이트합니다. 자세한 내용은 Amazon Inspector SBOM 생성기의 이전 버전을 참조하세요.	2025년 11월 4일

플러그인 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인이 CVE-2025-58188 및 CVE-2025-61725 에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인식합니다. Amazon Inspector SSM 플러그인이 이러한 CVE의 영향을 받지 않는 것으로 확인되었으며 이 감지를 해결하기 위해 업데이트가 배포되었습니다.	2025년 11월 3일
플러그인 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인이 CVE-2025-47907 에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인식합니다. Amazon Inspector SSM 플러그인이 이러한 CVE의 영향을 받지 않는 것으로 확인되었으며 이 감지를 해결하기 위해 업데이트가 배포되었습니다.	2025년 8월 8일
새 정책	Amazon Inspector는 Amazon Inspector에 대한 전체 액세스 권한과 기타 관련 서비스에 대한 액세스를 제공하는 새로운 관리형 정책을 추가합니다. 자세한 내용은 Amazon Inspector를 위한AWS 관리형 정책을 참조하세요.	2025년 7월 3일
업데이트된 기능	이제 새로운 AWS 리전에서 Amazon Inspector를 사용할 수 있습니다. 자세한 내용은 리전 및 엔드포인트 섹션을 참조하세요.	2025년 7월 1일

업데이트된 기능	Amazon Inspector에서 종결된 조사 결과의 보존 기간을 업데이트했습니다. Amazon Inspector는 연결된 리소스가 삭제, 종료되거나 더 이상 스캔할 수 없는 경우 3일 후에 조사 결과를 제거합니다. 자세한 내용은 Amazon Inspector 조사 결과 이해를 참조하세요.	2025년 6월 25일
업데이트된 기능	Amazon Inspector는 Amazon EC2 스캔 및 Amazon ECR 스캔을 위해 지원되는 운영 체제를 업데이트합니다. Amazon EC2 스캔은 이제 Fedora 버전 42 및 Ubuntu 버전 25.04를 지원합니다. Amazon ECR 스캔은 이제 Alpine 버전 3.22, Fedora 버전 42 및 Ubuntu 버전 25.04를 지원합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2025년 6월 18일
새로운 특성	Amazon Inspector는 이제 자사 애플리케이션 소스 코드, 타사 애플리케이션 종속성 및 코드형 인프라에서 취약성을 스캔합니다. 자세한 내용은 Amazon Inspector 코드 보안을 참조하세요.	2025년 6월 17일

플러그인 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인이 CVE-2025-0913 및 CVE-2025-4673 에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인식합니다. Amazon Inspector SSM 플러그인이 이러한 CVE의 영향을 받지 않는 것으로 확인되었으며 이 감지를 해결하기 위해 업데이트가 배포되었습니다.	2025년 6월 13일
새로운 특성	Amazon Inspector는 이제 활발하게 사용된 컨테이너 이미지와 클러스터에서 컨테이너 이미지가 마지막으로 사용된 시기를 표시할 수 있습니다. 자세한 내용은 실행 중인 컨테이너에 컨테이너 이미지 매핑을 참조하세요.	2025년 5월 16일
지원되는 운영 체제 업데이트	Amazon Inspector는 BusyBox에 대한 지원을 추가합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2025년 5월 13일
업데이트된 정책	Amazon Inspector는 AmazonInspector2ServiceRolePolicy 라는 서비스 연결 역할에 새 권한을 추가합니다. 이 권한을 통해 IP 주소 및 인터넷 게이트웨이를 설명할 수 있습니다. 자세한 내용은 Amazon Inspector를 위한 AWS 관리형 정책을 참조하세요.	2025년 4월 29일

플러그인 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인이 CVE-2025-22871 에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인식합니다. Amazon Inspector SSM 플러그인이 이러한 CVE의 영향을 받지 않는 것으로 확인되었으며 이 감지를 해결하기 위해 업데이트가 배포되었습니다.	2025년 4월 21일
플러그인 업데이트	Amazon Inspector는 Amazon Inspector SSM 플러그인이 CVE-2020-8911 , CVE-2020-8912 및 CVE-2024-45337 에 대한 취약성 조사 결과를 생성할 수 있는 시나리오를 인식합니다. Amazon Inspector는 이러한 CVE의 영향을 받지 않는 것으로 확인되었으며 이 감지를 해결하기 위해 업데이트가 배포되었습니다.	2025년 4월 18일
Amazon Inspector SBOM 생성기 업데이트 장	Amazon Inspector는 Amazon Inspector SBOM 생성기 버전을 업데이트합니다. 자세한 내용은 Amazon Inspector SBOM 생성기의 이전 버전을 참조하세요.	2025년 4월 16일

Amazon Inspector SBOM 생성기 업데이트 장	Amazon Inspector는 Amazon Inspector SBOM 생성기 장에 새 주제를 추가합니다. 이 주제에서는 Sbomgen가 소프트웨어 재료표에서 라이선스 정보를 추적하는 방법을 설명합니다. 자세한 내용은 Amazon Inspector SBOM 생성기 라이선스 컬렉션을 참조하세요.	2025년 4월 16일
관리형 정책으로 업데이트	Amazon Inspector는 Amazon ECS 및 Amazon EKS 작업에 대한 읽기 전용 액세스를 허용하는 권한을 추가합니다. 자세한 내용은 Amazon Inspector에 대한 서비스 연결 역할 권한을 참조하세요.	2025년 3월 25일
지원되는 운영 체제 업데이트	Amazon Inspector는 Amazon EC2 및 Amazon ECR 스캔의 일부로 더 이상 SUSE Linux Enterprise Server 12.5를 지원하지 않습니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2025년 3월 21일
지원되는 운영 체제 업데이트	Amazon Inspector는 Amazon ECR 스캔에 Chainguard 및 Wolfi에 대한 지원을 추가합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2025년 3월 21일

목차 업데이트	Amazon Inspector는 Amazon Inspector 리소스 태그 지정에 대한 장을 추가합니다. 자세한 내용은 Amazon Inspector 리소스에 태그 지정을 참조하세요.	2025년 2월 25일
목차 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기 장에 새 주제를 추가합니다. 자세한 내용은 Amazon Inspector SBOM 생성기 포괄적인 운영 체제 컬렉션을 참조하세요.	2025년 1월 28일
업데이트된 기능	Amazon Inspector는 Lambda 표준 스캔을 위해 지원되는 런타임 목록에 nodejs202.x 및 python3.13를 추가합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2025년 1월 24일
업데이트된 기능	Amazon Inspector는 Amazon EC2 및 Amazon ECR에 대해 지원되는 운영 체제 목록에서 Oracle Linux (Oracle) 7 및 SUSE Linux Enterprise Server (SLES) 15.5를 제거합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2024년 12월 31일

업데이트된 기능	Amazon Inspector는 Amazon EC2 및 Amazon ECR에 지원되는 운영 체제 목록에 Ubuntu 24.10을 추가합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2024년 12월 12일
목차 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기 장에 새 주제를 추가합니다. 자세한 내용은 Amazon Inspector SBOM 생성기를 참조하세요.	2024년 12월 9일
업데이트된 기능	Amazon Inspector는 `amazon:inspector:sbom_generator` 테이블을 업데이트하여 네임스페이스를 추가 및 제거합니다. 자세한 내용은 Amazon Inspector와 함께 CycloneDX 네임스페이스 사용을 참조하세요.	2024년 12월 9일
업데이트된 기능	Amazon Inspector는 CodePipeline을 사용한 스캔 작업을 지원하도록 CI/CD 통합 특성을 업데이트합니다. 자세한 내용은 CodePipeline에서 Amazon Inspector 스캔 작업 사용을 참조하세요.	2024년 11월 26일
목차 업데이트	Amazon Inspector는 Amazon Inspector SBOM 생성기에 대한 장을 포함하도록 목차를 재구성합니다. 자세한 내용은 Amazon Inspector SBOM 생성기를 참조하세요.	2024년 11월 22일

업데이트된 기능	Amazon Inspector는 Amazon EC2 및 Amazon ECR에 대해 지원되는 운영 체제 목록에서 Fedora 39를 제거합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2024년 11월 22일
업데이트된 기능	Amazon Inspector는 Amazon ECR에 대해 지원되는 운영 체제 목록에서 Alpine 3.17을 제거합니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2024년 11월 22일
업데이트된 기능	Amazon Inspector는 Amazon Inspector SBOM 생성기의 이전 버전에 Sbomgen 버전을 추가합니다.	2024년 11월 19일
업데이트된 기능	Amazon Inspector에서 지원되는 런타임으로 AL2가 추가됩니다. 자세한 내용은 Amazon Inspector에서 지원하는 운영 체제 및 프로그래밍 언어를 참조하세요.	2024년 8월 26일
업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 명령문을 추가했습니다. 새 명령문을 사용하면 Amazon Inspector가 AWS Lambda에서 함수 태그를 반환할 수 있습니다.	2024년 7월 31일

업데이트된 기능	Amazon Inspector에서 새로운 보안 제어 기능을 출시했습니다. 자세한 내용은 AWS Security Hub CSPM 사용 설명서에서 Amazon Inspector 제어를 참조하세요.	2024년 7월 11일
업데이트된 기능	Amazon Inspector SBOM 생성기가 이제 Dockerfiles 및 Docker 컨테이너 이미지를 스캔하여 보안 취약성을 유발할 수 있는 잘못된 구성이 있는지 확인합니다. 자세한 내용은 Amazon Inspector Dockerfile 검사를 참조하세요.	2024년 6월 10일
업데이트된 기능	CodeCatalyst 작업을 지원하도록 Amazon Inspector의 CI/CD 통합 특성이 업데이트되어 CodeCatalyst 워크플로에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 자세한 내용은 CodeCatalyst 작업 사용을 참조하세요.	2024년 6월 7일
업데이트된 기능	Amazon Inspector에 CIS 스캔 결과의 CSV 파일을 다운로드하는 옵션이 포함되었습니다. 자세한 내용은 Amazon EC2 인스턴스에 대한 Center for Internet Security(CIS) 스캔에서 CIS 스캔 결과 보기 및 다운로드를 참조하세요.	2024년 5월 3일

업데이트된 기능	GitHub Actions를 지원하도록 Amazon Inspector의 CI/CD 통합 특성이 업데이트되어 GitHub 워크플로에 Amazon Inspector 취약성 스캔을 추가할 수 있습니다. 자세한 내용은 GitHub Actions와 함께 Amazon Inspector 사용을 참조하세요.	2024년 4월 29일
업데이트된 기능	Amazon Inspector에서 관리형 정책 `AmazonInspector2FullAccess` 를 업데이트하여 서비스 연결 역할 `AWSServiceRoleForAmazonInspector2Agentless` 를 생성합니다. 이를 통해 사용자는 Amazon Inspector를 활성화할 때 에이전트 기반 스캔 및 에이전트 없는 스캔을 수행할 수 있습니다.	2024년 4월 24일
업데이트된 기능	Amazon Inspector에서 종결된 조사 결과의 보존 기간을 30일에서 7일로 업데이트했습니다. 자세한 내용은 Amazon Inspector의 조사 결과 이해를 참조하세요.	2024년 2월 12일
업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 명령문을 추가했습니다. 새 명령문을 사용하면 Amazon Inspector에서 인스턴스에 대한 CIS 스캔을 시작할 수 있습니다.	2024년 1월 23일

새 정책	Amazon Inspector에는 인스턴스 프로파일의 일부로 사용하여 인스턴스에 대한 CIS 스캔을 허용할 수 있는 새로운 관리형 정책인 AmazonInspector2ManagedCisPolicy 정책이 추가되었습니다.	2024년 1월 23일
새로운 기능	Amazon Inspector에서는 이제 컨테이너 이미지를 가져올 때 컨테이너 이미지의 ECR 재스캔 기간을 새로 고칩니다. 푸시 또는 가져오기 날짜를 기준으로 재스캔 기간을 변경하려면 ECR 재스캔 기간 구성을 참조하세요.	2024년 1월 23일
새로운 기능	Amazon Inspector에서는 이제 EC2 인스턴스에 대해 Center for Internet Security(CIS) 스캔을 실행할 수 있습니다. 자세한 내용은 Amazon Inspector CIS 스캔을 참조하세요.	2024년 1월 23일
새로운 기능	Amazon Inspector는 이제 CI/CD 파이프라인에서 컨테이너 이미지를 스캔할 수 있습니다. 자세한 내용은 CI/CD integration with Amazon Inspector를 참조하세요.	2023년 11월 30일

새 정책	Amazon Inspector는 Amazon Inspector가 에이전트 없는 스캔을 위해 EC2 인스턴스에서 Amazon EBS 스냅샷을 스캔할 수 있도록 허용하는 새 정책을 추가했습니다. 정책에 대한 자세한 내용은 에이전트 없는 스캔을 참조하세요.	2023년 11월 27일
새로운 특성	Amazon Inspector는 이제 에이전트 없는 스캔을 통해 SSM 에이전트 없이 지원되는 Linux Amazon EC2 인스턴스를 스캔할 수 있도록 지원합니다. 자세한 내용은 에이전트 없는 스캔을 참조하세요.	2023년 11월 27일
새로 지원되는 리소스	Amazon Inspector에서 이제 MacOS Amazon EC2 인스턴스 스캔을 지원합니다. 지원되는 macOS 버전은 지원되는 운영 체제: Amazon EC2 스캔을 참조하십시오.	2023년 10월 5일
새로운 리전	이제 아시아 태평양(자카르타), 아프리카(케이프타운), 아시아 태평양(오사카), 유럽(취리히)에서 Amazon Inspector를 사용할 수 있습니다.	2023년 9월 29일
새로운 특성	이제 제외 태그를 사용하여 Amazon Inspector 스캔에서 EC2 인스턴스를 제외할 수 있습니다.	2023년 9월 14일

새로운 특성	Amazon Inspector에서 Elastic Load Balancing 대상 그룹에 속하는 Amazon EC2 인스턴스의 네트워크 구성을 스캔할 수 있는 새로운 권한이 추가되었습니다.	2023년 8월 31일
새로운 특성	Amazon Inspector에서 이제 패키지 취약성 조사 결과에 대한 취약성 인텔리전스 세부 정보를 제공합니다.	2023년 7월 31일
업데이트된 기능	Amazon Inspector에서 읽기 전용 사용자가 리소스에 대한 Software Bill of Materials (SBOM)를 내보낼 수 있는 새로운 권한을 추가했습니다.	2023년 6월 29일
새로운 특성	이제 Amazon Inspector에서 스캔 중인 리소스에 대해 SBOM을 내보낼 수 있습니다.	2023년 6월 13일
새로운 특성	Lambda 코드 스캔이 이제 완전히 일반 공개되었습니다. Lambda 코드 스캔 조사 결과에서 식별된 코드를 암호화할 수 있는 새로운 특성이 추가되었습니다. 또한 Lambda 코드 스캐닝은 이제 코드에 대한 권장 수정 재작성을 제공합니다.	2023년 6월 13일

업데이트된 기능	Amazon Inspector에서 AmazonInspector2ReadOnlyAccess 정책에 새 명령문을 추가했습니다. 새 명령문은 읽기 전용 사용자가 자신의 계정에 대한 Lambda 코드 스캔 상태 및 조사 결과에 대한 세부 정보를 검색할 수 있도록 허용합니다.	2023년 5월 2일
새로운 특성	Amazon Inspector에서 특정 CVE를 처리하는지 확인할 수 있는 취약성 데이터베이스 검색 기능이 추가되었습니다.	2023년 5월 1일
업데이트된 기능	Amazon Inspector는 Lambda 스캔을 활성화할 때 Amazon Inspector가 계정에서 AWS CloudTrail 서비스 연결 채널을 생성할 수 있도록 허용하는 새 권한을 AmazonInspector2ServiceRolePolicy 정책에 추가했습니다. 이를 통해 Amazon Inspector는 사용자 계정의 CloudTrail 이벤트를 모니터링할 수 있습니다.	2023년 4월 30일
업데이트된 기능	Amazon Inspector에서 AmazonInspector2FullAccess 정책에 새 명령문을 추가했습니다. 이 명령문은 사용자가 Lambda 코드 스캔에서 코드 취약성 조사 결과에 대한 세부 정보를 검색할 수 있도록 허용합니다.	2023년 4월 17일

업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 명령문을 추가했습니다. 새 명령문은 Amazon Inspector에서 Amazon EC2 심층 검사를 위해 정의한 사용자 지정 경로에 대한 정보를 Amazon EC2 Systems Manager로 보낼 수 있도록 허용합니다.	2023년 4월 17일
새로운 특성	Amazon Inspector는 Amazon Inspector 심층 검사라는 Linux EC2 인스턴스에 대한 추가 지원을 제공하는데, 이 기능은 애플리케이션 프로그래밍 언어 패키지에 패키지 취약성이 있는지 인스턴스를 스캔합니다.	2023년 4월 17일
업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 명령문을 추가했습니다. 새로운 문을 통해 Amazon Inspector는 AWS Lambda 함수의 개발자 코드 스캔을 요청하고 Amazon CodeGuru Security로부터 스캔 데이터를 수신할 수 있습니다. 또한 Amazon Inspector에서 IAM 정책을 검토할 수 있는 권한이 추가되었습니다. Amazon Inspector는 이 정보를 사용하여 Lambda 함수의 코드 취약성을 스캔합니다.	2023년 2월 28일

새로운 특성	Amazon Inspector는 Lambda 코드 스캔이라는 Lambda 함수에 대한 추가 지원을 제공하는데, 이 기능은 Lambda 함수의 개발자 코드에서 보안 취약성을 스캔합니다.	2023년 2월 28일
업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 명령문을 추가했습니다. 이 명령문은 Amazon Inspector에서 AWS Lambda 함수가 마지막으로 간접적으로 호출된 시간에 대한 정보를 CloudWatch에서 검색할 수 있도록 허용합니다. 이 정보를 사용하여 사용자 환경에서 지난 90일 동안 활성화된 Lambda 함수에 초점을 맞추어 스캔을 수행할 수 있습니다.	2023년 2월 20일
업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 명령문을 추가했습니다. 새 명령문은 Amazon Inspector에서 AWS Lambda 함수에 대한 정보를 검색할 수 있도록 허용합니다. Amazon Inspector는 이 정보를 사용하여 Lambda 함수의 보안 취약성을 스캔합니다.	2022년 11월 28일
새로운 특성	Amazon Inspector에 스캔 AWS Lambda 함수에 대한 지원이 추가되었습니다.	2022년 11월 28일

콘텐츠 업데이트	Amazon Inspector에서 Amazon Simple Storage Service(S3) 버킷으로 조사 결과 보고서를 내보내는 절차, 정책 예제 및 팁이 추가되었습니다.	2022년 10월 14일
새로운 내용	Amazon Inspector 콘솔을 사용하여 AWS 환경의 Amazon Inspector 적용 범위를 평가하는 방법에 대한 정보가 추가되었습니다. Amazon Inspector이 정보에는 환경의 개별 리소스에 대한 상태 값 설명이 포함됩니다.	2022년 10월 7일
새로운 특성	Amazon Inspector에서 이제 패키지 취약성을 해결하는 방법에 대한 추가 세부 정보를 제공합니다. 조사 결과 세부 정보에 새로운 필드가 추가되었습니다. 새로운 필드는 패키지 업데이트를 통해 수정이 가능한지 여부에 대한 컨텍스트를 제공합니다. 수정이 가능한 경우 조사 결과의 제안된 해결 방법 섹션에 수정을 실행할 수 있는 명령이 표시됩니다.	2022년 9월 2일

업데이트된 기능	Amazon Inspector에서 AmazonInspector2ServiceRolePolicy 정책에 새 작업을 추가했습니다. 새 작업은 Amazon Inspector에서 SSM 연결 실행을 설명할 수 있도록 허용합니다. 또한 Amazon Inspector에서 `AmazonInspector2` 소유 SSM 문서와 SSM 연결을 생성, 업데이트, 삭제 및 시작할 수 있도록 리소스 범위가 추가되었습니다.	2022년 8월 31일
새로운 특성	Amazon Inspector에서 이제 Windows 인스턴스에 대한 스캔을 지원합니다. 지원되는 Windows 운영 체제를 실행하는 SSM 관리형 인스턴스를 Amazon Inspector에서 스캔할 수 있습니다. Windows 호스트 스캔은 Amazon Inspector SSM 플러그인을 통해 수행되며, 이 플러그인은 Amazon Inspector에서 자동으로 생성한 새 SSM 연결을 통해 설치되고 간접적으로 호출됩니다.	2022년 8월 31일
업데이트된 기능	Amazon Inspector는 Amazon Inspector가 다른 AWS 파티션에서 소프트웨어 인벤토리를 수집할 수 있도록 AmazonInspector2ServiceRolePolicy 정책의 리소스 범위를 업데이트했습니다.	2022년 8월 12일

업데이트된 기능	AmazonInspector2ServiceRolePolicy 정책에서 Amazon Inspector에서 SSM 연결을 생성, 삭제 및 업데이트할 수 있도록 작업의 리소스 범위가 재구성되었습니다.	2022년 8월 10일
새로운 특성	Amazon Inspector에서 이제 ECR 자동 재스캔 기간 설정의 변경을 지원합니다. Amazon ECR 자동 재스캔 기간 설정에 따라 Amazon Inspector에서 리포지토리로 푸시된 이미지를 지속적으로 모니터링하는 기간이 결정됩니다. 이미지가 스캔 기간보다 오래된 경우 Amazon Inspector에서 더 이상 이미지를 스캔하지 않고 이미지에 대한 기존 조사 결과를 모두 종결합니다. 모든 새 계정의 ECR 자동 재스캔 기간은 자동으로 전체 기간으로 설정됩니다. 이전에 생성한 계정의 ECR 자동 재스캔 기간은 30일이었지만 이제는 30일, 180일 또는 전체 스캔 기간 중에서 선택할 수 있습니다.	2022년 6월 25일
새로운 기능	Amazon Inspector 기능에 대한 읽기 전용 액세스를 허용하는 새로운 AWS 관리형 정책인 AmazonInspector2ReadOnlyAccess 정책을 Amazon Inspector했습니다.	2022년 1월 21일

정식 출시	이는 Amazon Inspector 사용 설명서의 최초 공개 릴리스입니다.	2021년 11월 29일

Amazon Inspector Security Research

Amazon Inspector는 NPM 레지스트리에서 악성 패키지를 지속적으로 모니터링하고 식별하여 공급망 공격으로부터 애플리케이션을 보호합니다.

최신 업데이트: 2026-02-06 12:00:00 UTC

감지 요약

- 수명 합계: 식별된 악성 패키지 191,801개
- 이번 달: 147개의 새로운 악성 패키지가 식별됨
- 지난 달: 527개의 새로운 악성 패키지가 식별됨
- 이번 주: 147개의 새로운 악성 패키지가 식별됨
- 지난 주: 96개의 새로운 악성 패키지가 식별됨

최근 악성 패키지 보고서(지난 10개)

패키지 이름	MAL-ID	감지 날짜
web3-sinon	MAL-2026-807	2026-02-06
web3-chain-sinon	MAL-2026-806	2026-02-06
정렬 배열	MAL-2026-805	2026-02-06
브레드크럼 서비스	MAL-2026-804	2026-02-06
@sbseg-plugin/qbo-web-app-ui	MAL-2026-802	2026-02-06
@rsgweb/utils	MAL-2026-801	2026-02-06
@rsgweb/tina	MAL-2026-800	2026-02-06

패키지 이름	MAL-ID	감지 날짜
@rsgweb/rockstar-account	MAL-2026-799	2026-02-06
@rsgweb/modules-core-www-page	MAL-2026-798	2026-02-06
@rsgweb/modules-core-feedback	MAL-2026-797	2026-02-06

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하십시오.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.