aws

사용 설명서

AWS 보안 에이전트

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

AWS 보안 에이전트: 사용 설명서

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

Table of Contents

소개 1
AWS Security Agent란 무엇입니까? 1
주요 기능 1
이점 2
AWS Security Agent 기능 3
AWS Security Agent 작동 방식 4
개요 4
리소스 계층 구조 및 수명 주기 이해 7
조직 전체에서 공유되는 내용 7
에이전트 공간당 범위 8
GitHub 리포지토리가 계층 구조에 맞는 방식 9
보안 기능의 주요 차이점 10
리소스 관계 이해 11
시작하기 13
에 가입 AWS 13
경로 선택 13
빠른 시작: 침투 테스트 실행 14
사전 조건 15
1단계: AWS 콘솔에서 AWS Security Agent 설정 15
2단계: 침투 테스트 활성화 및 도메인 확인 16
3단계: 소스 코드 연결(선택 사항) 16
4단계: 침투 테스트 생성 및 실행 17
5단계: 침투 테스트 결과 검토 18
빠른 시작: 코드 검토 실행 18
사전 조건 15
1단계: AWS 콘솔에서 AWS Security Agent 설정 15
2단계: 코드 검토 활성화 및 구성 19
3단계: 코드 검토 생성 및 실행 21
4단계: 코드 검토 결과 검토 21
빠른 시작: 위협 모델 실행 22
1단계: AWS 콘솔에서 AWS Security Agent 설정 15
2단계: 소스 코드 연결 23
3단계: 위협 모델 생성 및 실행 24
4단계: 위협 검토 24

관리자용(콘솔) 26
구성의 예 26
에이전트 스페이스 설정 및 생성 26
AWS Security Agent 설정 26
에이전트 스페이스 생성 31
통합 연결 33
Agent Spaces와의 통합 작동 방식 34
AWS Security Agent를 GitHub 리포지토리에 연결 35
GitHub 통합 제거 41
AWS Security Agent를 GitLab 리포지토리에 연결 43
AWS Security Agent를 GitLab 자체 관리형에 연결 46
GitLab 통합 제거 49
AWS Security Agent를 GitHub Enterprise Server에 연결 50
GitHub Enterprise Server 통합 제거 54
Atlassian 설치 ID 찾기 55
AWS Security Agent를 Bitbucket 리포지토리에 연결 55
Bitbucket 통합 제거 59
AWS Security Agent를 Confluence에 연결 60
Confluence 통합 제거 63
프라이빗 호스팅 소스 제어에 연결 64
에이전트를 프라이빗 VPC 리소스에 연결 69
기능 구성 72
침투 테스트 활성화 72
GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화 78
코드 검토 활성화 78
위협 모델링 활성화 85
사용자가 침투 테스트 및 코드 검토 결과의 수정을 시작할 수 있도록 지원 88
S3 버킷에서 에이전트 리소스 제공 89
침투 테스트를 위한 애플리케이션 도메인 활성화 90
침투 테스트에 사용되는 관리형 대상 도메인 92
보안 요구 사항 관리 94
보안 요구 사항 관리 94
AWS 관리형 팩 100
문서에서 보안 요구 사항 생성 102
요구 사항 생성을 위한 문서 준비 103
사용자 및 액세스 관리 105

사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한 부여 105
AWS Security Agent에 대한 IAM 역할 생성 107
고객 관리형 키 113
문제 해결 136
액세스 거부: 잘못된 GitHub 계정 유형이 선택되었거나 잘못된 조직 이름이 지정됨 136
액세스 거부: 조직에 GitHub 앱을 설치할 수 있는 권한 부족 136
침투 테스트 중에 에이전트가 엔드포인트에 연결할 수 없음 137
추가 도움말 받기 137
사용자 및 개발자용(웹 앱 및 IDE) 138
AWS Security Agent 웹 앱에 로그인 138
액세스 방법 138
IAM Identity Center(SSO)로 로그인 138
관리자 액세스(IAM)로 로그인 140
설계 검토 생성 141
사전 조건 15
1단계: 설계 검토 생성 시작 141
2단계: 설계 검토 이름 지정 142
3단계: 설계 파일 업로드 142
4단계: 설계 검토 시작 143
다음 단계 31
설계 검토의 결과 검토 143
위협 모델 생성 147
사전 조건 15
AWS Security Agent가 위협 모델을 실행하는 방법 148
위협 모델 페이지에 액세스 148
위협 모델 생성 149
위협 모델 실행 151
위협 모델 실행 모니터링 151
위협 모델 편집 152
위협 모델 삭제 153
다음 단계 31
위협 모델의 위협 검토 153
풀 요청에서 코드 보안 조사 결과 검토 158
풀 요청에서 코드 검토 작동 방식 159
코드 검토 결과 이해 159
보안 조사 결과에 대한 대응 160

코드 검토 조사 결과 필터링 160
다음 단계 31
코드 검토 생성 163
사전 조건 15
코드 검토 페이지에 액세스 163
코드 검토 생성 164
코드 검토 실행 169
코드 검토 실행 모니터링 169
다음 단계 31
코드 검토의 결과 검토 171
코드 검토 결과 수정 177
S3를 사용하여 차등 코드 스캔 실행 179
차등 스캔 작동 방식 180
사전 조건 15
1단계: diff 파일 생성 181
2단계: S3에 차이 업로드 181
3단계: 차등 코드 검토 작업 시작 181
4단계: 스캔 모니터링 182
5단계: 조사 결과 검토 183
할당량 및 제한 183
다음 단계 31
IDE에서 코드 보안 스캔 실행 184
IDE 통합 작동 방식 184
사전 조건 15
MCP 서버 설치 185
최초 설정 186
전체 보안 스캔 실행 187
차등 스캔 실행 188
위협 모델 검토 실행 188
조사 결과 검토 189
자동 수정 사항 적용 189
자동 스캔 제안(Kiro) 190
사용 가능한 스캔 유형 190
환경 변수 191
문제 해결 191
할당량 및 제한 183

다음 단계 31
침투 테스트 생성 192
사전 조건 15
침투 테스트 생성 시작 193
침투 테스트 이름 지정 193
침투 테스트 범위 구성 194
IAM 역할 구성 196
자동 코드 수정 178
VPC 리소스 구성(선택 사항) 197
인증 자격 증명 구성(선택 사항) 199
추가 리소스 연결(선택 사항) 201
침투 테스트 생성 204
침투 테스트 결과 검토 205
침투 테스트를 위한 인증 자격 증명 제공 215
침투 테스트 결과 수정 220
보안 및 참조 222
보안 222
보안 고려 사항 222
AWS 관리형 정책 229
데이터 보호 232
ID 및 액세스 관리 235
인시던트 대응 249
규정 준수 확인 250
복원력 251
인프라 보안 252
인터페이스 VPC 엔드포인트 253
구성 및 취약성 분석 257
교차 서비스 혼동된 대리자 방지 257
보안 모범 사례 258
IAM 작업 및 리소스 마이그레이션 261
CloudTrail을 사용하여 로깅 265
Service Quotas 268
작업 할당량 268
구성 할당량 268
문서 기록 269
..... cclxxii

소개

AWS Security Agent가 수행하는 작업, 작동 방식 및 리소스가 서로 어떻게 연결되는지 알아봅니다.

AWS Security Agent란 무엇입니까?

AWS Security Agent는 개발 수명 주기 동안 애플리케이션을 사전에 보호하는 프론티어 에이전트입니다. 조직의 요구 사항에 맞는 자동화된 보안 검토를 수행하고, 위협 모델을 구축하여 애플리케이션이 공격될 수 있는 방법을 식별하고, 온디맨드로 컨텍스트 인식 침투 테스트를 제공합니다. AWS Security Agent는 설계부터 배포까지 보안을 지속적으로 검증하여 모든 환경에서 취약성을 조기에 방지하는 데 도움이 됩니다.

보안 팀은 AWS 콘솔에서 승인된 권한 부여 라이브러리, 로깅 표준 및 데이터 액세스 정책 등 조직 보안 요구 사항을 한 번 정의합니다. AWS Security Agent는 개발 전반에 걸쳐 이러한 보안 요구 사항을 자동으로 적용하여 표준에 따라 아키텍처 문서 및 코드를 평가하고 위반을 감지할 때 특정 지침을 제공합니다. 이를 통해 모든 팀에서 설계 및 코드 검토를 위한 일관된 보안이 적용됩니다.

배포 검증을 위해 AWS Security Agent는 침투 테스트를 주기적 병목 현상에서 온디맨드 기능으로 변환합니다. 보안 팀은 대상 URLs, 인증 세부 정보, 소스 코드 및 애플리케이션 설명서를 제공합니다. AWS Security Agent는 심층적인 애플리케이션 이해를 개발하고 정교한 공격 체인을 실행하여 취약성을 발견하고 검증하므로 팀이 필요할 때마다 테스트할 수 있습니다.

주요 기능

AWS Security Agent는 전체 개발 수명 주기에 걸쳐 포괄적인 보안 기능을 제공합니다.

보안 검토 설계

AWS Security Agent는 코드가 작성되기 전에 설계 문서에 대한 실시간 보안 피드백을 제공하고 조직 보안 요구 사항 준수를 평가하여 보안을 왼쪽으로 전환합니다. 보안 팀은 웹 애플리케이션을 통해 문서를 업로드하고 조사 결과의 우선순위를 지정하여 시간이 많이 걸리는 수동 검토를 집중 분석으로 가속화하는 해결 지침을 받습니다. 모든 설계 검토에 보안 표준을 사전에 포함하면 후기 아키텍처 재작업을 줄이고 여러 개발 팀과 보조를 맞출 수 있습니다.

위협 모델링

AWS Security Agent는 애플리케이션의 위협 모델을 구축하여 공격 방법을 식별합니다. 기술적 설계 문서(범위 문서)를 제공하여 에이전트가 집중하는 내용을 정의하거나, 소스 코드를 에이전트가 기존 시스템을 이해할 수 있는 컨텍스트로 제공하거나, 둘 다 제공합니다. AWS Security Agent는 심각도 수준

및 실행 가능한 권장 사항과 함께 STRIDE 범주(스푸핑, 변조, 거부, 정보 공개, 서비스 거부, 권한 상승)별로 분류된 일련의 위협과 함께 애플리케이션의 아키텍처, 구성 요소, 신뢰 경계, 데이터 흐름 및 보안 태세를 설명하는 시스템 개요를 생성합니다. 위협 모델은 코드와 설계가 발전함에 따라 다시 실행할 수 있는 재사용 가능한 구성으로, 개발 수명 주기 전반에 걸쳐 반복적인 보안 평가를 가능하게 합니다.

코드 보안 검토

AWS Security Agent는 두 가지 보완 접근 방식을 통해 애플리케이션을 사전에 보호합니다. 먼저 웹 애플리케이션에서 코드 검토를 생성하여 GitHub, GitLab, Bitbucket 또는 GitHub Enterprise Server 리포지토리 또는 S3 버킷의 전체 소스 코드를 포괄적으로 스캔하여 보안 취약성을 식별하고 전체 코드베이스에서 조직 요구 사항 준수를 검증할 수 있습니다. 둘째, AWS Security Agent가 코드 변경 사항을 검토하고 보안 조사 결과를 풀 요청 또는 병합 요청 설명으로 직접 게시하는 연결된 리포지토리에 대한 자동 풀 요청 분석을 활성화할 수 있습니다. 두 경우 모두 개발자는 문제 해결 지침을 받으며 AWS Security Agent는 풀 요청을 자동으로 생성하거나 식별된 취약성에 대한 코드 수정과 요청을 병합할 수 있습니다. 이를 통해 모든 리포지토리에 대한 보안 전문 지식이 포함되어 개발 파이프라인의 보안 관련 지연을 줄이고 모든 코드베이스에서 평가를 확장할 수 있습니다.

침투 테스트

AWS Security Agent는 맞춤형 다단계 공격 시나리오를 통해 보안 취약성을 검색, 검증, 보고 및 해결하기 위해 특수 AI 에이전트를 배포하여 온디맨드 침투 테스트를 제공합니다. 에이전트는 소스 코드와 설명서를 분석하여 자동화된 보안 스캔 도구가 찾을 수 없는 취약성을 식별하고 악용하여 애플리케이션의 컨텍스트를 이해합니다. 영향 분석, 재현 가능한 공격 경로로 조사 결과를 문서화하고 ready-to-implement 코드 수정으로 풀 요청을 생성하여 주기적인 평가를 중요한 것으로만 제한되지 않고 모든 애플리케이션에서 확장되는 지속적 검증으로 변환합니다.

이점

온디맨드 접근성

AWS Security Agent는 보안 전문 지식에 대한 즉각적인 액세스를 제공합니다. 개발 팀은 필요할 때마다 온디맨드 방식으로 설계 검토, 위협 모델, 코드 분석 및 침투 테스트를 실행하여 최신 개발 주기의 속도를 일치시키고 모든 단계에서 선제적 보안을 활성화할 수 있습니다.

조직 요구 사항 자동 검증

AWS 콘솔에서 조직의 보안 요구 사항을 한 번 정의합니다. AWS Security Agent는 모든 설계 및 코드 보안 검토 중에 모든 애플리케이션에서 이러한 요구 사항을 자동으로 검증하여 팀이 일반 체크리스트가 아닌 특정 요구 사항을 해결할 수 있도록 합니다.

보안 전문 지식 확장

AWS Security Agent는 조직 보안 요구 사항의 적용을 자동화하여 개발 속도와 일치하도록 보안 검토를 확장합니다. AWS 콘솔을 통해 요구 사항을 중앙에서 구성하고, 조사 결과 분석을 통해 포괄적인 검토를 수행하고, 조직 전체의 침투 테스트 범위를 관리합니다.

확인된 취약성에 대한 실행 가능한 수정 사항

AWS Security Agent는 증명 기반 악용을 통해 침투 테스트 중에 발견된 보안 조사 결과를 검증하여 재현 가능한 악용 경로와 포괄적인 영향 분석을 제공하고 개발자 친화적인 언어로 ready-to-implement할 수 있는 수정 사항을 사용하여 풀 요청을 생성합니다. 이를 통해 팀은 오탐에 시간을 낭비하지 않고 합법적인 고위험 보안 위험에 집중할 수 있습니다.

다중 및 하이브리드 클라우드 지원

AWS Security Agent는 AWS, 온프레미스, 하이브리드, 멀티클라우드 및 SaaS 환경에서 운영되므로 인프라 또는 플랫폼 선택에 관계없이 일관된 보안 지침과 테스트를 보장합니다.

AWS Security Agent 기능

AWS Security Agent는 개발 수명 주기 전반에 걸쳐 네 가지 핵심 보안 기능을 제공합니다.

- 설계 보안 검토 - 설계 및 아키텍처 문서를 검토하여 보안 위험을 식별합니다. 웹 애플리케이션을 통해 문서를 업로드하면 서비스가 조직의 보안 요구 사항에 따라 문서를 분석합니다. AWS Security Agent는 승인된 권한 부여 라이브러리, 로깅 표준 및 데이터 액세스 정책과 같은 정의된 보안 요구 사항의 규정 준수를 평가합니다. 이렇게 하면 개발 프로세스 초기에 안전하지 않은 설계 및 정책 위반을 포착할 수 있습니다.
- 위협 모델링 - 애플리케이션의 위협 모델을 구축하여 공격 방법을 식별합니다. 설계 문서를 범위 문서로 제공하여 분석의 포커스를 정의하거나, 소스 코드를 에이전트가 기존 시스템을 이해할 수 있는 컨텍스트로 정의하거나, 두 가지 모두를 정의합니다. AWS Security Agent는 심각도 수준 및 실행 가능한 권장 사항과 함께 STRIDE 범주(스푸핑, 변조, 거부, 정보 공개, 서비스 거부, 권한 상승)로 분류된 일련의 위협과 함께 애플리케이션의 아키텍처, 구성 요소, 신뢰 경계, 데이터 흐름 및 보안 태세를 설명하는 시스템 개요를 생성합니다. 각 위협은 소스 코드의 증거로 다시 연결됩니다.
- 코드 보안 검토 - 리포지토리 및 S3 소스의 코드를 분석하여 언어, 프레임워크 및 아키텍처 전반의 보안 취약성과 조직 보안 요구 사항 위반을 식별합니다. 웹 애플리케이션에서 코드 검토를 생성하여 전체 소스 코드를 포괄적으로 스캔하거나 풀 요청 설명을 활성화하여 GitHub에서 코드 변경 사항을 자동으로 검토할 수 있습니다. AWS Security Agent는 특정 문제 해결 지침을 제공하며 식별된 취약성

에 대한 코드 수정을 통해 풀 요청을 자동으로 생성할 수 있습니다. 이렇게 하면 모든 개발 팀에서 보안 정책을 일관되게 적용할 수 있습니다.

- 온디맨드 침투 테스트 - 맞춤형 다단계 공격 시나리오를 통해 라이브 웹 애플리케이션 및 APIs의 보안 취약성을 검색, 검증, 보고 및 해결합니다. 테스트 범위, 인증 세부 정보 및 리소스를 지정하여 웹 애플리케이션을 통해 Pentest를 생성하도록 서비스를 구성합니다. AWS Security Agent는 제공된 소스 코드 및 설명서에서 애플리케이션 컨텍스트를 개발하고 정교한 공격 체인을 실행하여 정적 분석 및 기존 도구가 놓치는 악용 가능한 취약성을 식별합니다. 또한 ready-to-implement 있는 코드 수정을 제공하고 풀 요청을 코드 리포지토리에 직접 생성하여 취약성을 더 빠르게 해결할 수 있습니다.

AWS Security Agent 작동 방식

AWS Security Agent는 세 가지 인터페이스에서 작동하여 개발 수명 주기 전반에 걸쳐 선제적 애플리케이션 보안을 제공합니다. 보안 구성은 AWS Management Console에서 관리되고, 보안 검토는 Security Agent 웹 애플리케이션에서 수행되며, 자동 코드 및 보안 결과 수정은 GitHub와 같은 코드 리포지토리 플랫폼에서 직접 수행됩니다.

개요

AWS Security Agent는 세 가지 주요 구성 요소로 구성됩니다.

- AWS Management Console - 에이전트 공간 구성, 보안 요구 사항 정의, 침투 테스트 구성, 코드 리포지토리 연결, 사용자 액세스 관리
- Security Agent 웹 애플리케이션 - 설계 검토 수행, 위협 모델 생성 및 실행, 침투 테스트 실행, 코드 검토 생성 및 실행, 할당된 Agent Spaces 내에서 보안 조사 결과 검토
- 코드 리포지토리 통합 - 풀 요청 및 침투 테스트 수정 풀 요청에 대한 자동 코드 검토를 받습니다. 현재 AWS Security Agent는 GitHub에 대한 연결을 지원합니다.

사용하는 인터페이스로 식별할 수 있는 세 가지 역할 중 하나로 AWS Security Agent를 사용합니다.

Role	작업 위치 및 작업
관리자	AWS 관리 콘솔에서 작동 - 에이전트 스페이스를 설정하고, 보안 요구 사항을 정의하고, 기능을 구성하고, 사용자 액세스를 관리합니다.
User	Security Agent 웹 애플리케이션에서 작동 - 설계 검토, 위협 모델, 침투 테스트 및 코드 검토를 실행하고 결과 결과를 검토합니다.

Role	작업 위치 및 작업
개발자	GitHub 또는 IDE(예: Kiro 또는 Claude Code)에서 작동 - 풀 요청에 대한 자동 보안 결과를 수신하고 개발 환경을 벗어나지 않고 코드 스캔을 실행합니다.

전체적으로 이러한 역할 이름을 사용하여 각 작업을 수행하는 사용자를 나타냅니다. 한 사람이 둘 이상의 역할을 보유할 수 있습니다.

콘솔 구성

관리자는 AWS Management Console을 통해 AWS Security Agent를 구성합니다.

에이전트 스페이스 - AWS 관리 콘솔에서 첫 번째 에이전트 스페이스를 생성하면 AWS Security Agent가 계정에 대한 웹 애플리케이션을 생성합니다. 생성하는 각 에이전트 스페이스는 보호하려는 고유한 애플리케이션 또는 프로젝트를 나타냅니다. 웹 애플리케이션에서 사용자는 보안 평가를 수행할 때 사용할 에이전트 공간을 선택합니다.

보안 요구 사항 - AWS Security Agent가 설계 및 코드 검토 중에 평가하는 보안 요구 사항을 보안 요구 사항 팩으로 구성합니다. 업계 표준에 따라 AWS 관리형 팩을 활성화하거나, 조직의 정책에 맞는 사용자 지정 팩을 생성하거나, 보안 설명서를 업로드하여 요구 사항을 생성합니다. 요구 사항은 모든 에이전트 스페이스에 적용됩니다.

침투 테스트 구성 - 다음을 통해 각 에이전트 스페이스에 대한 침투 테스트 기능을 구성합니다.

- DNS 또는 HTTP 확인을 통해 테스트할 대상 도메인 확인
- 프라이빗 애플리케이션 테스트를 위한 VPC 액세스 구성
- 침투 테스트 실행을 위한 CloudWatch 로깅 설정
- 테스트 자격 증명을 위한 AWS Secrets Manager 또는 Lambda 함수 구성
- 추가 애플리케이션 컨텍스트를 위한 S3 버킷 지정

코드 검토 구성 - 다음을 통해 각 에이전트 스페이스에 대한 코드 검토 기능을 구성합니다.

- 소스 코드가 포함된 GitHub 리포지토리 또는 S3 버킷 연결
- 코드 검토 설정 선택(보안 취약성, 사용자 지정 요구 사항 또는 둘 다)
- GitHub에서 코드 변경 사항의 자동 검토를 위한 풀 요청 설명 활성화
- 코드 검토 실행을 위한 CloudWatch 로깅 설정

위협 모델링 구성 - 다음을 통해 각 에이전트 스페이스에 대한 위협 모델링 기능을 구성합니다.

- 소스 코드가 포함된 소스 코드 리포지토리 또는 S3 버킷 연결
- 범위 문서(기능 설계 문서)를 추가하여 분석을 특정 기능에 집중
- 위협 모델 실행을 위한 CloudWatch 로깅 구성
- AWS 리소스 액세스를 위한 서비스 역할 설정

통합 - GitHub 리포지토리를 각 에이전트 스페이스에 연결하여 주요 기능을 활성화합니다.

- 보다 정확한 침투 테스트를 위한 애플리케이션 컨텍스트 제공
- 전체 리포지토리 스캔 및 풀 요청 분석을 위한 코드 검토 활성화
- 코드 검토 및 침투 테스트 결과에 대한 풀 요청을 통해 자동화된 코드 수정 활성화

사용자 액세스 - 사용자가 Security Agent 웹 애플리케이션에 액세스하는 방법을 관리합니다. IAM Identity Center(SSO)를 활성화한 경우 AWS Security Agent 콘솔에서 사용자를 할당하여 웹 애플리케이션에 대한 직접 SSO 액세스를 제공합니다. IAM 전용 액세스를 사용하는 경우 AWS 콘솔 액세스 권한이 있는 사용자는 모든 에이전트 스페이스에 대해 콘솔의 관리자 액세스 링크를 통해 웹 애플리케이션을 시작할 수 있습니다.

웹 애플리케이션 활동

사용자는 Security Agent 웹 애플리케이션에 액세스하여 할당된 에이전트 스페이스 내에서 보안 평가를 수행합니다.

에이전트 공간 선택 - 웹 애플리케이션에 로그인할 때 사용자는 작업할 에이전트 공간을 선택합니다. 사용자는 할당된 에이전트 공간만 보고 액세스할 수 있습니다.

설계 검토 - 조직 보안 요구 사항에 대한 분석을 위해 설계 문서 및 아키텍처 사양을 업로드합니다. 문제 해결 지침에 따라 조사 결과를 검토합니다.

위협 모델 - 소스 코드, 기술 설계 문서(범위 문서) 또는 둘 다를 제공하여 위협 모델을 생성하고 실행합니다. 범위 문서는 에이전트가 분석에 초점을 맞춘 내용을 정의합니다. 소스 코드는 기존 시스템에 대한 컨텍스트를 제공합니다. 각 실행은 심각도 수준 및 실행 가능한 권장 사항이 있는 STRIDE 범주로 분류된 위협 세트와 함께 애플리케이션의 아키텍처, 신뢰 경계, 데이터 흐름 및 보안 태세를 설명하는 시스템 개요를 생성합니다.

코드 검토 - 전체 소스 코드에서 포괄적인 정적 분석을 수행하는 코드 검토를 생성하고 실행합니다. GitHub 리포지토리 또는 S3 소스를 선택하고, 스캔 설정을 구성하고, 수정 지침에 따라 자세한 보안 조

사 결과를 검토합니다. AWS Security Agent는 보안 취약성을 식별하고 전체 코드베이스에서 조직의 사용자 지정 보안 요구 사항 준수를 검증합니다.

침투 테스트 - 대상 URLs, 인증 세부 정보 및 설명서를 제공하여 침투 테스트를 구성하고 실행합니다. AWS Security Agent는 다단계 공격 시나리오를 통해 악용 가능한 취약성을 발견하기 위해 자율 테스트를 수행합니다.

조사 결과 검토 - 영향 분석, 재현 가능한 공격 경로 및 문제 해결 지침을 포함하여 설계 검토, 위협 모델, 코드 검토 및 침투 테스트에서 자세한 보안 조사 결과를 검토합니다.

수정 사항 검증 - 문제 해결을 구현한 후 보안 평가를 다시 실행하여 취약성이 해결되었는지 확인합니다.

GitHub 통합

AWS Security Agent는 GitHub와 직접 통합되어 개발자의 워크플로에서 자동화된 보안 피드백을 제공합니다.

풀 요청 설명 - 관리자가 AWS Security Agent GitHub 앱을 설치하고 에이전트 스페이스에 대한 코드 검토 설명으로 코드 검토를 활성화하면 AWS Security Agent는 연결된 리포지토리의 풀 요청을 자동으로 분석합니다. 개발자는 풀 요청 설명에서 직접 보안 조사 결과 및 수정 지침을 받아 조직 보안 요구 사항 및 일반적인 취약성에 대한 코드 변경을 검증합니다.

자동 문제 해결 - 사용자가 코드 검토를 위해 자동 코드 문제 해결을 활성화하면 AWS Security Agent는 식별된 취약성에 대한 수정 사항을 생성하고 관련 GitHub 리포지토리에 풀 요청을 제출합니다.

침투 테스트 문제 해결 - 관리자가 콘솔에서 결과 문제 해결을 활성화하면 사용자는 웹 애플리케이션에서 침투 테스트 결과에 대한 자동 문제 해결을 요청할 수 있습니다. AWS Security Agent는 취약성을 해결하기 위해 코드 수정을 사용하여 연결된 GitHub 리포지토리에 대한 풀 요청을 엽니다.

리소스 계층 구조 및 수명 주기 이해

AWS Security Agent는 조직 전체에서 공유되는 내용과 애플리케이션당 범위가 결정되는 계층 구조로 보안 테스트 리소스를 구성합니다. 이 구조를 이해하면 AWS Security Agent를 효과적으로 구성하고 다양한 리소스를 찾고 관리할 위치를 파악하는 데 도움이 됩니다.

조직 전체에서 공유되는 내용

AWS Security Agent의 일부 리소스는 조직 수준에서 한 번 구성되며 모든 애플리케이션 및 에이전트 스페이스에 적용됩니다. 이러한 테넌트 수준 리소스는 일관성을 제공하고 중복 구성 작업을 줄입니다.

Resource	정의	공유되는 이유
보안 요구 사항	AWS Security Agent가 설계 및 코드 검토 중에 검증하는 내용을 정의하는 조직 보안 표준	보안 정책은 모든 애플리케이션에 적용됩니다. 이를 한 번 정의하면 AWS Security Agent가 모든 곳에서 적용합니다.
GitHub 통합	AWS Security Agent에 연결할 권한이 있는 등록된 GitHub 조직 또는 사용자 계정	GitHub 조직을 한 번 등록한 다음 필요에 따라 특정 리포지토리를 에이전트 스페이스에 연결합니다.
IAM Identity Center 구성	사용자가 AWS Security Agent에 액세스하는 방법을 제어하는 SSO 설정	중앙 집중식 ID 관리는 조직의 모든 에이전트 스페이스에 적용됩니다.

> **Important**
>
> 보안 요구 사항을 변경하면 모든 에이전트 스페이스에서 향후 모든 설계 검토 및 코드 검토에 영향을 미칩니다. 기존 검토는 영향을 받지 않습니다.

에이전트 공간당 범위

각 에이전트 스페이스는 보호하려는 고유한 애플리케이션 또는 프로젝트를 나타냅니다. 에이전트 공간 수준의 리소스는 특정 애플리케이션으로 범위가 지정되므로 팀마다 구성 및 평가를 독립적으로 수행할 수 있습니다.

Resource	정의	애플리케이션당 범위가 지정된 이유
침투 테스트 구성	애플리케이션 내의 특정 기능, API 엔드포인트 또는 기능에 대한 구성 테스트	각 애플리케이션에는 해당 애플리케이션과 관련된 고유한 대상, 인증 방법 및 범위 경계가 있습니다.
설계 검토	설계 문서의 개별 아키텍처 보안 평가	각 애플리케이션에는 독립적으로 평가되는 자체 아키텍처 및 설계 문서가 있습니다.

Resource	정의	애플리케이션당 범위가 지정된 이유
위협 모델	시스템 개요를 구축하고 소스 코드, 설계 문서 또는 둘 다에서 위협을 식별하는 위협 모델링 평가	각 애플리케이션에는 고유한 코드와 설계가 있으며 위협은 독립적으로 모델링됩니다. 위협 모델은 코드와 설계가 발전함에 따라 다시 실행할 수 있는 재사용 가능한 구성입니다.
통합	이 에이전트 스페이스에 연결된 소스 및 설명서 공급자(GitHub, GitLab, Bitbucket, GitHub Enterprise Server 및 Confluence)	애플리케이션마다 소스와 설명서가 다릅니다. 에이전트 스페이스 수준에서 연결하면 애플리케이션 경계가 명확해집니다.
코드 검토 설정	연결된 소스, 스캔 설정 및 PR 주석 활성화를 포함한 코드 검토 기능 구성	각 애플리케이션에는 자체 리포지토리와 보안 검토 요구 사항이 독립적으로 구성되어 있습니다.
침투 테스트 문제 해결 설정	침투 테스트 결과에 대한 자동 수정 풀 요청을 수신할 수 있는 연결된 리포지토리의 구성	팀은 애플리케이션의 워크플로에 따라 AWS Security Agent가 코드 변경을 제출할 수 있는 위치를 제어합니다.
사용자 할당	이 특정 에이전트 스페이스에 액세스할 수 있는 사용자	팀은 자신이 담당하는 애플리케이션에 대한 보안 평가만 보고 작업을 체계적이고 집중적으로 유지합니다.

> **Tip**
>
> 팀 간의 명확한 경계를 유지하고 보안 평가를 효과적으로 구성하려면 애플리케이션 또는 프로젝트당 하나의 에이전트 공간을 생성하는 것이 좋습니다.

GitHub 리포지토리가 계층 구조에 맞는 방식

GitHub 리포지토리는 조직 리소스를 특정 애플리케이션에 연결하는 다단계 프로세스를 통해 통합됩니다.

1. 테넌트 수준에서 등록 - GitHub 조직 또는 사용자 계정에 대한 AWS 보안 에이전트 GitHub 앱을 한 번 승인

2. 에이전트 스페이스 수준에서 연결 - 각 에이전트 스페이스에 연결할 특정 리포지토리 선택
3. 리포지토리당 사용량 구성 - 연결된 각 리포지토리에 대해 특정 기능을 활성화합니다.
    - 코드 검토 - 전체 소스 코드 스캔 및 자동 풀 요청 분석
    - 침투 테스트 컨텍스트 - 침투 테스트 중 소스 코드의 애플리케이션 이해
    - 자동 코드 수정 - 코드 검토 및 침투 테스트 결과를 위한 취약성 수정이 포함된 자동 풀 요청

각 리포지토리에서 서로 다른 기능이 활성화된 여러 에이전트 스페이스에 단일 리포지토리를 연결할 수 있습니다.

보안 기능의 주요 차이점

AWS Security Agent의 각 보안 기능은 보안 팀이 이를 사용하는 방식에 따라 다른 워크플로 모델을 따릅니다.

침투 테스트: 독립적인 실행으로 재사용 가능한 구성

침투 테스트는 반복적인 보안 테스트를 지원하는 configuration-and-run 모델을 사용합니다.

- 한 번 생성, 여러 번 실행 - 범위 경계, 인증 및 테스트 파라미터를 사용하여 특정 대상(API 엔드포인트, 기능 영역)에 대한 구성 정의
- 독립 실행 - 보안을 개선하면서 동일한 구성을 여러 번 실행합니다. 각 실행은 독립적이며 새로운 결과를 생성합니다.

이 모델은 개선 사항을 개발하고 배포할 때 지속적인 보안 검증을 지원합니다.

설계 검토: 복제를 사용한 일회성 평가

설계 검토는 재사용 가능한 구성 모델을 따르지 않는 독립적인 평가입니다.

- 단일 평가 - 각 설계 검토는 조직의 보안 요구 사항에 따라 업로드된 문서를 한 번 분석합니다.
- 다시 실행할 수 없음 - 설계 검토를 재사용할 수 없습니다. 동일한 검토를 다시 실행할 수 없습니다.
- 업데이트를 위한 복제 - 기존 설계 검토를 복제하여 사전 로드된 원본 문서로 새 검토를 생성하므로 문서를 업데이트하고 새 분석을 실행할 수 있습니다.

이 모델은 point-in-time 아키텍처 보안 평가를 지원합니다.

코드 검토: 온디맨드 스캔 및 자동 PR 분석을 통한 재사용 가능한 구성

코드 검토는 소스 코드를 보호하기 위한 두 가지 작업 모드를 제공합니다.

- 전체 코드 검토(웹 애플리케이션) - GitHub 리포지토리 또는 S3 소스를 선택하는 코드 검토 구성을 생성한 다음 온디맨드로 포괄적인 스캔을 실행합니다. 각 실행은 전체 소스 코드에서 정적 분석을 수행하고 문제 해결 지침에 따라 결과를 생성합니다. 코드가 발전할 때와 동일한 코드 검토 구성을 다시 실행할 수 있습니다.
- 풀 요청 설명(GitHub) - 연결된 GitHub 리포지토리에 대한 자동 분석을 활성화합니다. AWS Security Agent는 풀 요청이 검토 준비 완료로 표시되면 자동으로 검토하고 보안 조사 결과를 GitHub에 직접 주석으로 게시합니다.

두 모드 모두 구성된 코드 검토 설정(보안 취약성, 사용자 지정 요구 사항 또는 둘 다)을 사용하고 풀 요청을 통한 자동 코드 수정을 지원합니다.

위협 모델: 온디맨드 실행을 통한 재사용 가능한 구성

위협 모델은 아키텍처에 대한 반복적인 평가를 지원하는 configuration-and-run 모델을 사용합니다.

- 한 번 생성, 여러 번 실행 - 소스 코드를 소스로 선택하거나 설계 문서를 범위 문서로 업로드하거나 둘 다로 위협 모델을 정의합니다. 온디맨드로 실행하고 코드와 설계가 발전함에 따라 다시 실행합니다.
- 유연한 입력 - 소스 코드에만 위협 모델을 실행하거나 문서만 설계하거나 둘 다 실행합니다. 범위 문서는 에이전트가 분석에 초점을 맞춘 내용을 정의합니다. 소스 코드는 기존 시스템에 대한 컨텍스트를 제공합니다.
- 시스템 개요 및 위협 - 각 실행은 심각도, 증거 및 실행 가능한 권장 사항이 포함된 STRIDE 범주로 분류된 위협 세트와 함께 애플리케이션의 아키텍처, 신뢰 경계, 데이터 흐름 및 보안 태세를 설명하는 시스템 개요를 생성합니다.

리소스 관계 이해

계층 구조는 다양한 리소스를 구성하고 액세스하는 위치를 결정합니다.

AWS 관리 콘솔에서:

- 테넌트 수준 리소스 구성(보안 요구 사항, GitHub 통합, IAM Identity Center)
- 에이전트 스페이스 생성 및 관리

- 에이전트 공간 설정 구성(연결된 리포지토리, 코드 검토 활성화, 침투 테스트 수정)

Security Agent 웹 애플리케이션에서:

- 침투 테스트 구성 및 테스트 실행 생성 및 관리
- 설계 검토 생성 및 관리
- 연결된 리포지토리 및 S3 소스에 대한 코드 검토 생성, 관리 및 실행
- 소스 코드, 범위 문서 또는 둘 다에 대한 위협 모델 생성, 관리 및 실행
- 침투 테스트, 코드 검토, 설계 검토 및 위협 모델의 조사 결과 보기

GitHub에서:

- 풀 요청 코드 검토 조사 결과를 풀 요청 설명으로 보기
- 코드 검토 및 침투 테스트 결과에 대한 자동 문제 해결 풀 요청 수신(에이전트 스페이스에서 활성화된 경우)

> **Note**
>
> 풀 요청 코드 검토 결과는 GitHub에 표시됩니다. 전체 코드 검토, 침투 테스트 및 설계 검토 조사 결과는 Security Agent 웹 애플리케이션에 표시됩니다.

시작하기

가입하고 목표에 적합한 시작점을 찾은 다음 빠른 시작으로 첫 번째 평가를 실행합니다.

AWS 계정 가입

시작하려면 AWS 계정이 AWS필요합니다. AWS 계정 생성에 대한 자세한 내용은 AWS 계정 관리 참조 안내서의 AWS 계정 시작하기를 참조하세요.

경로 선택

AWS Security Agent로 수행할 작업에 적합한 시작점을 찾습니다. 아래 표를 사용하여 목표를 작업과 일치시킨 다음 해당 페이지로 바로 이동합니다. AWS Security Agent를 처음 사용하는 경우 the section called “AWS Security Agent란 무엇입니까?” 먼저 읽어 간단한 개요를 확인하세요.

...하고 싶습니다.	누가	여기에서 시작
설정하기 전에 AWS Security Agent가 수행하는 작업 이해	모든 사람	the section called “AWS Security Agent란 무엇입니까?”
서비스 설정 및 에이전트 공간 생성	관리자	the section called “AWS Security Agent 설정”
라이브 또는 배포된 애플리케이션의 악용 가능한 취약성 테스트	User	the section called “빠른 시작: 침투 테스트 실행”
소스 코드에서 취약성 및 정책 위반 여부를 스캔합니다.	User	the section called “빠른 시작: 코드 검토 실행”
애플리케이션이 공격될 수 있는 방법 모델링(STRIDE)	User	the section called “빠른 시작: 위협 모델 실행”
코드를 작성하기 전에 설계에 대한 보안 피드백 받기	User	the section called “설계 검토 생성”
내 IDE(Kiro 또는 Claude Code)를 벗어나지 않고 코드 스캔	개발자	the section called “IDE에서 코드 보안 스캔 실행”

...하고 싶습니다.	누가	여기에서 시작
병합하기 전에 변경된 줄만 스캔	개발자	the section called “S3를 사용하여 차등 코드 스캔 실행”
풀 요청 및 병합 요청에 대한 자동 보안 설명 가져오기	관리자	the section called “코드 검토 활성화”
조사 결과를 검토하고 먼저 수정할 사항을 결정합니다.	User	침투 테스트, 코드 검토, 위협 모델, 설계 검토

> **Note**
>
> AWS Security Agent는 작업하는 인터페이스로 식별할 수 있는 세 가지 역할을 사용합니다. Admin은 AWS Management Console(설정 및 구성), User는 웹 애플리케이션(평가 실행 및 결과 검토), Developer는 GitHub 또는 Kiro 또는 Claude Code와 같은 IDE에서 작동합니다. 한 사람이 둘 이상의 역할을 보유할 수 있습니다. 전체 정의는 섹션을 참조하세요the section called “AWS Security Agent 작동 방식”.

빠른 시작: 침투 테스트 실행

이 빠른 시작은 AWS Security Agent를 사용하여 첫 번째 침투 테스트(펜테스트)를 실행하는 방법을 안내합니다. 침투 테스트는 확인된 대상 도메인에 대해 배포된 애플리케이션을 연습하고 심각도 등급 및 지원 증거가 있는 보안 조사 결과를 반환합니다. 공개적으로 액세스할 수 있는 애플리케이션을 다룹니다. 프라이빗 VPC에서 호스팅되는 애플리케이션을 테스트하려면 섹션을 참조하세요the section called “침투 테스트 활성화”.

> **Note**
>
> AWS Security Agent를 설정하고 테스트 범위를 정의하려면 AWS Management Console에 대한 액세스 권한이 필요하며, 침투 테스트를 생성하고 실행하려면 웹 애플리케이션에 대한 액세스 권한이 필요합니다.

사전 조건

시작하기 전에, 다음 사항을 확인해야 합니다.

- AWS Security Agent를 설정할 수 있는 권한이 있는 AWS Management Console에 액세스합니다.
- 테스트하려는 애플리케이션을 호스팅하는 라이브 대상 도메인입니다.
- 소유권을 확인할 수 있도록 해당 도메인에 대한 DNS 레코드 또는 동일한 AWS 계정의 Route 53 호스팅 영역을 추가하는 기능입니다.
- (선택 사항) 에이전트에게 더 많은 컨텍스트를 제공하기 위한 애플리케이션의 소스 코드 리포지토리(GitHub, GitLab 또는 Bitbucket).

1단계: AWS 콘솔에서 AWS Security Agent 설정

AWS Security Agent를 아직 설정하지 않은 경우 초기 설정을 완료합니다.

1. AWS Management Console에서 AWS Security Agent로 이동합니다.
2. AWS Security Agent 설정을 선택합니다.
3. 에이전트 스페이스를 생성합니다. 에이전트 공간은 여러 사용자가 사용할 수 있으며 테스트하려는 모든 애플리케이션에 고유해야 합니다. 이름 및 설명을 입력합니다. 이름은 침투 테스트하려는 애플리케이션을 식별해야 합니다.
4. 사용자 액세스 구성에서 IAM 전용 액세스를 선택합니다.
   - 이 빠른 시작은 사용자가 AWS 콘솔에서 직접 웹 애플리케이션에 액세스할 수 있도록 하는 IAM Identity Center를 통한 Single Sign-On(SSO) 활성화를 다루지 않습니다.
   - AWS Management Console이 없는 사용자가 침투 테스트를 시작하도록 하려면 IAM Identity Center 통합을 활성화합니다. 자세한 내용은 the section called “사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한 부여”을 참조하세요.
5. AWS Security Agent 설정을 선택합니다.

> **Note**
>
> 설정을 선택하면 AWS Security Agent는 에이전트 스페이스를 생성하고 사용자가 침투 테스트, 코드 검토, 위협 모델 및 설계 검토를 실행할 수 있는 웹 애플리케이션을 설정합니다.

2단계: 침투 테스트 활성화 및 도메인 확인

> **Note**
>
> AWS 콘솔에서 테스트할 수 있는 항목의 범위를 정의합니다. 그런 다음 사용자는 웹 애플리케이션에서 해당 범위 내에서 특정 침투 테스트를 실행합니다.

1. 왼쪽 사이드바에서 에이전트 공간을 선택한 다음 1단계에서 생성한 에이전트 공간을 선택합니다.
2. 침투 테스트 탭을 선택한 다음 침투 테스트 설정을 선택하여 마법사를 엽니다.
3. 도메인 구성 - 테스트할 대상 도메인을 입력하고 확인 방법, DNS TXT 레코드 또는 HTTP 경로를 선택합니다. 도메인은 라이브 상태여야 하며 테스트하려는 애플리케이션을 호스팅해야 합니다. 다음을 선택합니다.
4. 도메인 확인 - 대상 도메인 테이블에서 각 도메인의 소유권을 확인합니다. AWS Security Agent는 확인된 도메인에 대해서만 테스트를 실행합니다. 자세한 단계와 복사할 정확한 값은 섹션을 참조하세요the section called “침투 테스트를 위한 애플리케이션 도메인 활성화”.
   - 동일한 AWS 계정의 Route 53 도메인 - 도메인을 선택하고 원클릭 확인을 선택합니다. AWS Security Agent가 DNS 레코드를 생성하고 확인을 완료합니다.
   - DNS TXT 레코드(기타 DNS 공급자) - 대상 도메인 테이블에서 레코드 이름 및 레코드 값을 복사하고 DNS 공급자와 함께 TXT 레코드로 추가한 다음 도메인을 선택하고 확인을 선택합니다. DNS 변경 사항이 전파되는 데 시간이 걸릴 수 있으므로 확인이 즉시 완료되지 않을 수 있습니다.
   - HTTP 경로 - 웹 서버에서 `.well-known/aws/securityagent-domain-verification.json`에 파일을 생성하고 형식의 확인 토큰을 추가한 `{"tokens": ["<token>"]}`다음 도메인을 선택하고 확인을 선택합니다.
5. (선택 사항) 추가 기능 구성 - CloudWatch 로그 그룹 및 자격 증명과 같은 선택적 리소스를 추가합니다. 서비스 액세스 섹션은 미리 구성되어 있습니다. AWS Security Agent는 기존 IAM 역할을 선택하지 않는 한 필요한 권한이 있는 서비스 역할을 생성합니다.

3단계: 소스 코드 연결(선택 사항)

소스 코드 공급자를 연결하면 애플리케이션에 대한 AWS Security Agent 컨텍스트가 제공되고 침투 테스트 적용 범위가 개선됩니다. 이 단계는 선택 사항입니다.

1. 침투 테스트 탭에서 페이지 상단의 침투 테스트를 위한 소스 코드 공급자 연결 배너에서 추가를 선택합니다.

2. 공급자를 등록하고 연결한 다음 침투 테스트에 사용할 리포지토리를 선택합니다.

전체 통합 흐름은 GitHub 리포지토리에 AWS Security Agent 연결 또는 공급자의 연결 주제를 참조하세요.

4단계: 침투 테스트 생성 및 실행

> **Note**
>
> AWS Security Agent 웹 애플리케이션에서 침투 테스트를 생성하고 실행합니다. 프로덕션을 거의 미러링하는 테스트 환경에 대해 테스트를 실행하는 것이 좋습니다.

1. 웹 애플리케이션을 시작합니다. 웹 앱 탭을 선택한 다음 관리자 액세스를 선택합니다.
2. 왼쪽 사이드바에서 침투 테스트를 선택한 다음 침투 테스트 생성을 선택합니다.
3. 침투 테스트 세부 정보 - 테스트 범위 및 로그 소스를 설정합니다.
   a. Pentest 이름을 입력합니다.
   b. 대상 URLs에서 테스트할 확인된 도메인(예: `https://example.com`)을 선택하거나 입력합니다. 확인된 도메인만 테스트할 수 있으며 하위 도메인은 자동으로 포함됩니다.
   c. (선택 사항) 액세스 가능한 URLs에서 에이전트가 로그인 및 탐색을 위해 연결해야 하지만 대상 도메인 외부의 자격 증명 공급자, CDNs 또는 APIs와 같이 공격해서는 안 되는 도메인을 추가합니다. AWS Security Agent는 이러한 도메인에 도달할 수 있지만 테스트하지는 않습니다. 대상 도메인만 공격됩니다.
   d. 네트워킹 구성 규칙을 검토하여 테스트 중에 트래픽을 수신할 수 있는 엔드포인트와 수신할 수 없는 엔드포인트를 확인합니다.
   e. 권한에서 서비스 역할을 선택하고 선택적으로 CloudWatch 로그 그룹을 선택합니다. 다음을 선택합니다.
4. (선택 사항) VPC 리소스 - 대상이 공개적으로 연결할 수 없는 프라이빗 네트워크에 있는 경우 VPC를 구성합니다. the section called “침투 테스트 활성화”을(를) 참조하세요.
5. (선택 사항) 인증 리소스 - 에이전트가 인증된 비공개 경로에 도달할 수 있도록 자격 증명을 제공합니다. 로그인 뒤의 적용 범위를 넓히고 취약성을 표시하려면 이를 추가하는 것이 좋습니다.
6. (선택 사항) 추가 구성 - 파일, GitHub 리포지토리 또는 S3 소스와 같은 애플리케이션 컨텍스트를 추가하여 결과 품질을 개선합니다. 자동 코드 수정을 활성화하고 여기에서 다른 실행 옵션을 설정할 수도 있습니다.

7. 지금 테스트를 시작하려면 생성 및 실행을 선택하고, 저장하고 나중에 실행하려면 pentest 생성을 선택합니다.

5단계: 침투 테스트 결과 검토

1. 침투 테스트를 완료하는 데 몇 시간이 걸릴 수 있습니다. 대부분의는 애플리케이션의 크기와 복잡성에 따라 16시간 이내에 완료됩니다.
2. 실행은 테스트 시작 전에 설정을 검증하는 Preflight 단계로 시작됩니다. 즉, 로깅 인프라를 설정하고, 대상 엔드포인트에 연결할 수 있는지 확인하고, 테스트 환경을 준비합니다. 사전 검사가 실패하면(예: 해결할 수 없는 대상 도메인) 테스트가 시작되기 전에 실행이 중지됩니다. Preflight 탭을 열어 실패한 검사를 확인하고 해결한 다음 새 실행을 시작합니다.
3. 실행이 완료되면 실행을 열고 실행 개요, 로그 및 결과 탭을 검토합니다.
4. 조사 결과 탭에서 조사 결과를 선택하여 설명, 심각도, 위험 유형 및 근거 증거를 확인합니다.

자세한 내용은 the section called “침투 테스트 생성” 및 the section called “침투 테스트 결과 검토” 섹션을 참조하세요.

빠른 시작: 코드 검토 실행

이 빠른 시작에서는 AWS Security Agent를 사용하여 첫 번째 코드 검토를 실행하는 방법을 안내합니다. AWS Security Agent는 소스 코드 리포지토리에서 보안 취약성을 검사하고 조직의 보안 요구 사항을 준수합니다.

> **Note**
>
> AWS Security Agent를 설정하려면 AWS Management Console에 액세스하고 코드 검토를 생성하고 실행하려면 웹 애플리케이션에 액세스해야 합니다.

사전 조건

시작하기 전에, 다음 사항을 확인해야 합니다.

- AWS Security Agent를 설정할 수 있는 권한이 있는 AWS Management Console에 액세스합니다.
- 소스 코드 리포지토리(GitHub, GitLab 또는 Bitbucket) 또는 검토하려는 코드가 포함된 Amazon S3 소스입니다.

1단계: AWS 콘솔에서 AWS Security Agent 설정

AWS Security Agent를 아직 설정하지 않은 경우 초기 설정을 완료합니다.

1. AWS Management Console에서 AWS Security Agent로 이동합니다.
2. AWS Security Agent 설정을 선택합니다.
3. 에이전트 스페이스를 생성합니다. 에이전트 공간은 여러 사용자가 사용할 수 있으며 테스트하려는 모든 애플리케이션에 고유해야 합니다. 첫 번째 에이전트 공간의 이름과 설명을 입력합니다. 이 이름은 웹 애플리케이션의 사용자에게 표시됩니다. 이름은 코드를 검토하려는 애플리케이션을 식별해야 합니다.
4. 사용자 액세스 구성에서 IAM 전용 액세스를 선택합니다.
   - 이 빠른 시작은 IAM Identity Center에서 Single Sign-On(SSO) 활성화를 다루지 않습니다. 이를 통해 사용자는 AWS 콘솔에서 AWS Security Agent 웹 애플리케이션에 직접 액세스할 수 있습니다.
   - AWS Management Console이 없는 사용자가 코드 검토를 실행하도록 하려면 IAM Identity Center 통합을 활성화합니다. 자세한 내용은 the section called “사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한 부여”을 참조하세요.
5. AWS Security Agent 설정을 선택합니다.

> **Note**
>
> 설정을 선택하면 AWS Security Agent는 에이전트 스페이스를 생성하고 사용자가 침투 테스트, 코드 검토, 위협 모델 및 설계 검토를 실행할 수 있는 웹 애플리케이션을 설정합니다.

2단계: 코드 검토 활성화 및 구성

> **Note**
>
> 이미 GitHub 리포지토리 또는 S3 버킷이 에이전트 스페이스에 연결되어 있는 경우(예: 침투 테스트 설정을 통해) 코드 검토가 이미 활성화되어 있습니다. 이 단계를 건너뛰고 웹 애플리케이션으로 바로 이동할 수 있습니다.

코드 검토 설정 마법사 열기

1. 왼쪽 사이드바에서 에이전트 공간을 선택한 다음 에이전트 공간을 선택합니다.
2. 헤더 또는 코드 검토 탭에서 코드 검토 활성화를 선택합니다.

통합, 리포지토리 및 버킷 연결

1. (GitHub 통합이 아직 없는 경우) GitHub 등록을 생성합니다. 이미 있는 경우 다음 단계로 건너뜁니다.
   a. 연결된 통합 섹션에서 추가를 선택한 다음 새 등록 생성을 선택합니다.
   b. GitHub를 선택하고 다음을 선택합니다.
   c. 설치 및 권한 부여를 선택한 다음 GitHub에서 설치를 완료합니다.
      i. 검토하려는 리포지토리를 소유한 GitHub 사용자 또는 조직을 선택합니다.
      ii. 모든 리포지토리를 선택하거나 리포지토리만 선택합니다.
      iii. 설치 및 승인을 선택하고 GitHub 인증을 완료합니다.
   d. AWS 관리 콘솔로 돌아가서 등록 이름을 입력하고 GitHub 앱을 설치한 계정 유형과 일치하는지 확인합니다.
   e. 연결을 선택하여 등록을 저장합니다.

   전체 GitHub 통합 흐름은 GitHub 리포지토리에 AWS 보안 에이전트 연결을 참조하세요.
2. GitHub 리포지토리를 연결합니다. 연결된 통합 섹션에서 추가를 선택한 다음 GitHub 등록을 선택합니다. 2단계 Connect GitHub 마법사가 열립니다.
   a. GitHub 리포지토리 연결에서 포함할 리포지토리를 선택하고 다음을 선택합니다.
   b. 기능 관리에서 리포지토리당 다음을 전환합니다.
      - 코드 검토 설명 - AWS Security Agent가 리포지토리의 풀 요청에 대한 설명으로 보안 조사 결과를 게시하도록 합니다.
      - 자동 문제 해결 - AWS Security Agent 웹 애플리케이션 요청이 조사 결과를 수정하는 요청을 가져오도록 허용합니다.
   c. 저장을 선택하여 설정 마법사로 돌아갑니다.
3. (선택 사항) S3 소스를 연결합니다. S3 버킷 섹션에서 S3 리소스 추가를 선택하고 소스 코드가 포함된 버킷의 S3 URI를 입력하거나 찾아보기를 선택하여 하나를 선택합니다.
4. 코드 검토 설정을 선택합니다. 기본 보안 요구 사항 및 취약성 조사 결과는 활성화한 보안 요구 사항과 일반적인 취약성을 모두 준수하는지 코드를 분석합니다.

5. 다음을 선택합니다.

구성 옵션

1. 선택적 CloudWatch 로그 그룹 및 서비스 액세스를 구성합니다. 기본 서비스 역할은 필요한 권한으로 미리 구성되어 있습니다.
2. 저장을 선택합니다.

3단계: 코드 검토 생성 및 실행

> **Note**
>
> AWS Security Agent 웹 애플리케이션에서만 코드 검토를 생성하고 실행합니다.

1. 웹 앱 탭을 선택한 다음 관리자 액세스를 선택하여 AWS Security Agent 웹 애플리케이션을 시작합니다.
2. 왼쪽 사이드바에서 코드 검토를 선택합니다.
3. 코드 검토 생성을 선택합니다.
4. 코드 검토를 구성합니다.
   a. 이 검토의 범위를 식별하는 제목을 입력합니다(예: "billing-service-security-review").
   b. 소스에서 2단계에서 에이전트 스페이스에 연결한 GitHub 리포지토리를 선택하거나 스캔하려는 S3 소스를 입력합니다.
   c. 구성된 역할에서 서비스 역할을 선택합니다.
   d. (선택 사항) AWS Security Agent가 모든 결과에 대한 수정 사항이 포함된 풀 요청을 자동으로 제출하도록 하려면 자동 코드 수정 활성화를 선택합니다.
5. 코드 검토 생성을 선택합니다.
6. 코드 검토 세부 정보 페이지에서 검토 시작을 선택합니다.

4단계: 코드 검토 결과 검토

1. 코드베이스의 크기에 따라 코드 검토에는 일반적으로 30~60분이 소요됩니다.
2. 실행은 스캔이 시작되기 전에 설정을 검증하는 Preflight 단계로 시작됩니다. AWS Security Agent가 리포지토리 또는 S3 소스에서 소스 코드를 가져올 수 있는지 확인하고 스캔 환경을 설정합니다. 사

전 검사가 실패하면(예: 소스에 액세스할 수 없음) 정적 분석 전에 실행이 중지됩니다. Preflight 탭을 열어 실패한 검사를 확인하고 해결한 다음 새 실행을 시작합니다.

3. 완료되면 완료된 실행으로 이동하여 결과 탭을 선택합니다.
4. list-detail 보기에서 조사 결과를 검토합니다.
   a. 왼쪽 패널에서 결과를 선택하여 세부 정보를 봅니다.
   b. 설명, 코드 위치, 증거 및 권장 수정 사항 섹션을 검토합니다.
   c. 수정 코드를 사용하여 수정 사항이 포함된 풀 요청을 생성하거나 해당 옵션을 활성화한 경우 자동 수정 PRs 검토합니다.

자세한 내용은 the section called “코드 검토 생성” 및 the section called “코드 검토의 결과 검토” 섹션을 참조하세요.

빠른 시작: 위협 모델 실행

이 빠른 시작은 AWS Security Agent를 사용하여 첫 번째 위협 모델을 실행하는 방법을 안내합니다. 위협 모델은 애플리케이션의 아키텍처를 분석하고 시스템 개요(AWS Security Agent가 시스템을 이해하는 방법)와 일련의 위협(각 공격 방법, 심각도 수준, STRIDE 분류 및 권장 사항 포함)을 생성합니다. 설계 문서(범위 문서)에서 위협 모델을 실행하여 포커스를 정의하거나, 소스 코드(소스)를 정의하여 기존 시스템에 대한 컨텍스트를 제공하거나, 둘 다 제공할 수 있습니다.

> **Note**
>
> AWS Security Agent를 설정하려면 AWS Management Console에 액세스하고 위협 모델을 생성하고 실행하려면 웹 애플리케이션에 액세스해야 합니다.

1단계: AWS 콘솔에서 AWS Security Agent 설정

AWS Security Agent를 아직 설정하지 않은 경우 초기 설정을 완료합니다.

1. AWS Management Console에서 AWS Security Agent로 이동합니다.
2. AWS Security Agent 설정을 선택합니다.
3. 에이전트 스페이스를 생성합니다. 에이전트 공간은 여러 사용자가 사용할 수 있으며 보호하려는 모든 애플리케이션에 고유해야 합니다. 첫 번째 에이전트 공간의 이름과 설명을 입력합니다. 이름은 위협 모델로 사용할 애플리케이션을 식별해야 합니다.

4. 사용자 액세스 구성에서 IAM 전용 액세스를 선택합니다.
   - 이 빠른 시작은 IAM Identity Center에서 Single Sign-On(SSO) 활성화를 다루지 않습니다. 이를 통해 사용자는 AWS 콘솔에서 AWS Security Agent 웹 애플리케이션에 직접 액세스할 수 있습니다.
   - AWS Management Console 액세스 권한이 없는 사용자가 위협 모델 시작과 같은 작업을 수행할 수 있도록 하려면 IAM Identity Center 통합을 활성화해야 합니다.
5. AWS Security Agent 설정을 선택합니다.

> **Note**
>
> 설정을 선택하면 AWS Security Agent는 에이전트 스페이스를 생성하고 사용자가 침투 테스트, 코드 검토, 위협 모델 및 설계 검토를 실행할 수 있는 웹 애플리케이션을 설정합니다.

2단계: 소스 코드 연결

> **Note**
>
> 에이전트 스페이스에 이미 연결된 리포지토리 또는 S3 버킷이 있는 경우(예: 코드 검토 또는 침투 테스트 설정을 통해)이 단계를 건너뛰고 웹 애플리케이션으로 직접 이동할 수 있습니다. 소스 코드를 연결하지 않고도 업로드된 범위 문서에서 항상 위협 모델을 실행할 수 있습니다.

에이전트가 기존 시스템을 이해하기 위한 컨텍스트로 사용할 소스 코드가 포함된 리포지토리 또는 S3 버킷을 연결합니다.

1. 왼쪽 사이드바에서 에이전트 공간을 선택한 다음 에이전트 공간을 선택합니다.
2. 통합 섹션으로 이동하여 소스 코드 공급자를 연결합니다.
3. 또는 소스 코드가 포함된 S3 버킷을 연결합니다.

전체 통합 흐름은 GitHub 리포지토리에 AWS 보안 에이전트 연결을 참조하세요.

3단계: 위협 모델 생성 및 실행

> **Note**
>
> AWS Security Agent 웹 애플리케이션에서 위협 모델을 생성하고 실행합니다.

1. AWS Management Console의 웹 앱 탭에서 웹 애플리케이션을 시작합니다. 또는 IAM Identity Center를 구성한 경우 직접 로그인합니다.
2. 왼쪽 사이드바에서 위협 모델을 선택합니다.
3. 위협 모델 생성을 선택합니다.
4. 위협 모델을 구성합니다.
   a. 이 위협 모델의 범위를 식별하는 제목을 입력합니다(예: "billing-service" 또는 "checkout-api").
   b. 하나 이상의 입력을 제공합니다.
      - 범위 문서에서 설계 문서(DOC, DOCX, JPEG, MD, PDF, PNG 또는 TXT)를 업로드하고 S3 URIs 입력하거나 Confluence 페이지를 선택합니다.
      - 소스에서 연결된 통합의 리포지토리를 선택하거나 소스 코드가 포함된 S3 URIs 입력합니다.

      > **Tip**
      >
      > 소스와 범위 문서를 모두 제공하여 위협 모델의 범위를 특정 설계(예: 기능 설계 문서)로 지정하는 동시에 에이전트에게 기존 시스템을 이해하기 위한 컨텍스트로 소스 코드를 제공합니다.

   c. 구성된 역할에서 서비스 역할을 선택합니다.
   d. (선택 사항) CloudWatch 로그에 대한 로그 그룹을 선택합니다.
5. 위협 모델 생성을 선택합니다.
6. 위협 모델 세부 정보 페이지에서 실행 시작을 선택합니다.

4단계: 위협 검토

1. 실행은 분석 작업을 진행합니다. 사전 처리 탭에서 진행 상황을 모니터링하고 로그 탭에서 작업 세부 정보를 볼 수 있습니다.
2. 완료되면 실행 상태가 완료됨으로 변경됩니다.

3. 개요 탭을 선택하여 시스템 개요 및 심각도 분포를 검토합니다.
4. 식별된 위협을 검토하려면 위협 탭을 선택합니다.
   a. 목록에서 위협을 선택하여 측면 패널에서 세부 정보를 봅니다.
   b. 문, 심각도, STRIDE 범주, 권장 사항, 증거 및 기타 필드를 검토합니다.
   c. 각 위협을 해결하거나 분류할 때 위협 상태를 해결됨 또는 무시됨으로 업데이트합니다.

자세한 내용은 the section called “위협 모델 생성” 및 the section called “위협 모델의 위협 검토” 섹션을 참조하세요.

관리자용(콘솔)

관리자는 AWS Management Console에서 AWS Security Agent를 설정하고 사용자가 AWS Security Agent 웹 애플리케이션을 통해 액세스하는 Agent Spaces를 구성합니다. 각 에이전트 스페이스는 특정 권한과 리소스가 있는 고유한 환경을 나타냅니다.

테스트하려는 각 애플리케이션에 대해 고유한 에이전트 공간을 생성하는 것이 좋습니다. 예를 들어 결제 애플리케이션과 작업 추적 애플리케이션이라는 두 개의 내부 프로젝트가 있는 경우 두 개의 개별 에이전트 스페이스를 생성해야 합니다.

구성의 예

관리자가 에이전트 공간을 설정하여 내부 결제 애플리케이션의 보안을 평가하는 것이 좋습니다. 관리자는 다음을 수행합니다.

- 도메인 확인(예: `beta.billing.example.com`)
- GitHub에 연결하고 코드 검토를 활성화합니다.
- 침투 테스트를 위한 적절한 VPC, 서브넷 및 보안 그룹을 할당하여 네트워크 액세스 구성

사용자가 침투 테스트 또는 설계 검토를 시작하면 미리 구성된 리소스 중에서 선택하여 정의한 가드레일 내에서 작업하면서 특정 평가 요구 사항에 대한 유연성을 유지할 수 있습니다.

에이전트 스페이스 설정 및 생성

조직에 대한 AWS Security Agent를 설정하고 각 애플리케이션의 리소스 및 평가 범위를 지정하는 에이전트 스페이스를 생성합니다.

AWS Security Agent 설정

첫 번째 에이전트 스페이스를 생성하고 사용자가 웹 애플리케이션에 액세스하는 방법을 설정하여 조직에 맞게 AWS Security Agent를 구성합니다.

이 초기 설정을 통해 관리자는 AWS 콘솔에서 보안 기능을 구성하고 Security Agent 웹 애플리케이션을 통해 설계 검토 및 침투 테스트에 액세스할 수 있습니다. 이 일회성 설정 후 간소화된 프로세스로 추가 에이전트 스페이스를 생성할 수 있습니다.

개요

에이전트 공간 이해

에이전트 스페이스는 특정 애플리케이션 또는 프로젝트를 보호하기 위한 전용 워크스페이스입니다. 여기에는 해당 애플리케이션에 대한 모든 보안 검토, 선택적으로 연결된 GitHub 리포지토리, 침투 테스트 구성, 결과 및 조사 결과가 포함됩니다.

에이전트 스페이스를 사용하면 각 애플리케이션의 보안 평가를 별도로 유지하여 보안 작업을 구성할 수 있으므로 팀이 특정 애플리케이션에 집중할 수 있습니다. 명확한 경계를 유지하려면 애플리케이션 또는 프로젝트당 하나의 에이전트 공간을 생성하는 것이 좋습니다.

보안 요구 사항은 조직 수준에서 정의되며 모든 에이전트 스페이스에 적용되는 반면, 각 에이전트 스페이스는 자체 설계 문서, 코드 리포지토리, 침투 테스트 구성, 침투 테스트 결과 및 보안 조사 결과를 유지 관리합니다.

에이전트 스페이스에 포함된 항목

각 에이전트 스페이스에는 다음이 포함됩니다.

- 선택적으로 애플리케이션 또는 프로젝트와 연결된 GitHub 리포지토리
- 애플리케이션에 대한 이전 설계 검토
- 애플리케이션별 침투 테스트 구성 및 경계
- 침투 테스트 테스트 결과 및 보안 조사 결과

설정 중에 구성할 내용

이 최초 설정 중에 모든 에이전트 스페이스에 적용되는 조직 결정을 내려야 합니다.

- 액세스 방법 - 사용자가 Security Agent 웹 애플리케이션에 액세스하는 방법 선택(IAM Identity Center SSO 또는 AWS 콘솔을 통한 IAM 전용 액세스)
- 권한 - 웹 애플리케이션이 AWS 서비스에 액세스하는 데 사용하는 IAM 역할 구성
- 첫 번째 에이전트 공간 - 이름과 설명을 사용하여 초기 에이전트 공간 생성

> **Note**
>
> 이 설정을 완료한 후 추가 에이전트 스페이스를 생성하는 것이 더 간단합니다. 이름과 설명을 제공하면 됩니다. 후속 에이전트 스페이스 생성에 the section called “에이전트 스페이스 생성” 대한 자세한 내용은 섹션을 참조하세요.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Security Agent 리소스를 생성하고 관리할 수 있는 권한
- 조직의 자격 증명 관리 요구 사항 이해
- (선택 사항) IAM 역할을 생성하고 IAM Identity Center를 구성할 수 있는 권한이 있는 AWS 계정 (SSO 액세스를 사용하는 경우)
- (선택 사항) 사용자 지정 권한 구성을 사용하려는 경우 기존 IAM 역할

1단계: 첫 번째 에이전트 공간 생성

웹 애플리케이션의 사용자에게 표시될 첫 번째 에이전트 공간의 기본 속성을 정의합니다.

1. AWS Security Agent 콘솔 페이지에서 보안 에이전트 설정을 선택합니다.
2. 에이전트 스페이스 이름 필드에 에이전트 스페이스의 이름을 입력합니다.

> **Note**
>
> 에이전트 스페이스 이름은 웹 애플리케이션의 사용자에게 표시되며이 스페이스가 나타내는 애플리케이션 또는 프로젝트를 식별하는 데 도움이 됩니다.

3. (선택 사항) 설명 필드에 에이전트 공간 목적을 구분하는 데 도움이 되는 설명을 입력합니다.

> **Tip**
>
> 설명은 에이전트 스페이스의 목적을 구분하는 데 도움이 됩니다. "고객 포털 웹 애플리케이션", "결제 처리 마이크로서비스" 또는 "내부 분석 플랫폼"과 같이이 에이전트 스페이스가 보호할 특정 애플리케이션 또는 프로젝트를 설명하는 것이 좋습니다.

2단계: 액세스 방법 선택

사용자가 Security Agent 웹 애플리케이션에 액세스하는 방법을 선택합니다. IAM Identity Center를 통해 SSO 액세스를 활성화하거나 AWS 콘솔을 통해 액세스를 제공할지 선택합니다.

> **Note**
>
> IAM 전용 액세스를 선택하고 나중에 IAM Identity Center를 사용하려면 AWS Security Agent 설정을 삭제하고 설정 프로세스를 다시 완료해야 합니다.

1. 액세스 방법 섹션에서 다음 옵션 중 하나를 선택합니다.
   - IAM Identity Center(SSO) - IAM Identity Center를 통해 팀에 SSO 액세스를 활성화합니다. 이 옵션은 중앙 집중식 사용자 관리가 필요하고 사용자가 AWS 콘솔을 거치지 않고 웹 애플리케이션에 직접 액세스하도록 하려는 팀에 권장됩니다.
   - IAM 전용 액세스 - AWS 콘솔의 관리자 액세스 링크를 통해 액세스를 제공합니다. 이 옵션은 설정하기가 더 간단하며 콘솔 기반 액세스를 선호하거나 SSO 기능이 필요하지 않은 팀에 적합합니다.
2. IAM Identity Center(SSO)를 선택한 경우 아래 2a단계로 계속 진행합니다.
3. IAM 전용 액세스를 선택한 경우 3단계로 건너뜁니다.

2a단계: IAM Identity Center 구성(SSO 액세스만 해당)

IAM Identity Center(SSO)를 액세스 방법으로 선택한 경우에만 이 단계를 완료합니다.

1. IAM Identity Center에 연결 섹션에서 AWS 리전을 검토합니다.

> **Important**
>
> IAM Identity Center를 활성화한 리전과 동일한 리전에서 애플리케이션을 구성해야 합니다. 표시된 리전은 에이전트 스페이스가 생성될 리전입니다. 에이전트 스페이스를 생성하는 리전과 동일한 리전에서 IAM Identity Center를 활성화해야 합니다.

2. IAM Identity Center 섹션에서 다음 중 하나를 선택합니다.
   - 계정 인스턴스 생성을 선택하여 새 IAM Identity Center 계정 인스턴스를 생성합니다.
   - 조직 인스턴스가 이미 동일한 리전에 있는 경우 AWS Security Agent가 해당 인스턴스에 자동으로 연결합니다.

> **Note**
>
> AWS Security Agent가 IAM Identity Center에 연결되면 IAM Identity Center의 사용자를 애플리케이션에 할당하여 액세스를 관리할 수 있습니다.

3. Active Directory 또는 외부 자격 증명 공급자를 연결하는 경우 정보 알림을 검토하세요.

> **Important**
>
> Active Directory 또는 다른 자격 증명 소스의 사용자를 이미 관리하고 있고 해당 자격 증명 소스를 IAM Identity Center에 연결하려는 경우 설정을 취소하고 먼저 IAM Identity Center 콘솔로 이동합니다. AWS Security Agent를 설정하기 전에 연결하려는 자격 증명 소스를 선택합니다. 나중에 자격 증명 소스를 변경하면 기존 사용자 할당이 제거될 수 있습니다.

3단계: 권한 구성(선택 사항)

Security Agent 웹 애플리케이션이 AWS 서비스, APIs 및 계정에 액세스하는 데 사용하는 IAM 역할을 구성합니다.

1. 권한 구성 - 선택 사항 섹션을 찾습니다.
2. 섹션이 축소된 경우 권한 구성 섹션을 선택하여 확장합니다.
3. 다음 옵션 중 하나를 선택합니다.
   - 기본 역할 생성 - AWS Security Agent는 웹 애플리케이션에 필요한 권한을 가진 새 IAM 역할을 자동으로 생성합니다.
   - 다른 역할 사용 - 사용 가능한 옵션에서 기존 IAM 역할 선택
4. 역할 설명을 검토합니다.

> **Note**
>
> 웹 애플리케이션이 다른 AWS 서비스, APIs 및 계정에 액세스할 수 있도록 기본 IAM 역할이 생성됩니다. 역할에는 보안 평가 작업에 필요한 권한이 부여됩니다.

4단계: 설정 완료

필요한 모든 설정을 구성한 후 설정을 완료하여 첫 번째 에이전트 스페이스를 생성하고 보안 에이전트 웹 애플리케이션을 설정합니다.

1. 모든 구성 섹션을 검토하여 정확성을 확인합니다.
2. 설정을 선택합니다.
3. AWS Security Agent는 에이전트 스페이스를 생성하고, 웹 애플리케이션을 설정하고, 필요한 AWS 리소스를 구성합니다.

> **Note**
>
> 초기 설정 후에는 침투 테스트 경계, 보안 요구 사항 및 GitHub 통합을 포함한 기능을 구성할 수 있습니다.

다음 단계

AWS Security Agent를 설정한 후:

- 조직의 보안 요구 사항 정의
- 에이전트 스페이스에 대한 도메인 확인 및 VPC 액세스를 포함한 침투 테스트 기능 구성
- 코드 검토 및 침투 테스트 컨텍스트를 위해 GitHub 리포지토리 연결
- (IAM Identity Center를 사용하는 경우) AWS Security Agent 콘솔의 에이전트 스페이스에 사용자 할당
- (IAM 전용 액세스를 사용하는 경우) AWS 콘솔의 관리자 액세스 링크를 통해 웹 애플리케이션을 시작합니다.
- 다른 애플리케이션을 위한 추가 에이전트 공간 생성(참조the section called “에이전트 스페이스 생성”)

에이전트 스페이스 생성

에이전트 스페이스를 생성하여 조직의 애플리케이션 또는 프로젝트를 보호합니다. 각 에이전트 스페이스는 해당 애플리케이션 또는 프로젝트와 관련된 보안 평가, 조사 결과 및 구성을 위한 워크스페이스를 제공하며 여러 사용자가 액세스할 수 있습니다.

이 절차에서는 초기 AWS Security Agent 설정을 이미 완료했다고 가정합니다. AWS Security Agent를 아직 설정하지 않은 경우 섹션을 참조하세요the section called “AWS Security Agent 설정”.

개요

에이전트 공간 이해

에이전트 스페이스는 특정 애플리케이션 또는 프로젝트를 보호하기 위한 전용 워크스페이스입니다. 여기에는 해당 애플리케이션에 대한 모든 보안 검토, 선택적으로 연결된 코드 리포지토리, 침투 테스트 구성, 결과 및 조사 결과가 포함됩니다.

에이전트 스페이스를 사용하면 각 애플리케이션의 보안 평가를 별도로 유지하여 보안 작업을 구성할 수 있으므로 팀이 특정 애플리케이션에 집중할 수 있습니다. 명확한 경계를 유지하려면 애플리케이션 또는 프로젝트당 하나의 에이전트 공간을 생성하는 것이 좋습니다.

에이전트 공간 및 에이전트 공간이 조직의 보안 구조에 어떻게 맞는지에 대한 포괄적인 설명은 섹션을 참조하세요the section called “리소스 계층 구조 및 수명 주기 이해”.

에이전트 스페이스에 포함된 항목

각 에이전트 스페이스에는 다음이 포함됩니다.

- 선택적으로 애플리케이션 또는 프로젝트와 연결된 코드 리포지토리
- 애플리케이션 또는 프로젝트에 대한 이전 설계 검토
- 애플리케이션별 침투 테스트 구성 및 경계
- 침투 테스트 테스트 결과 및 보안 조사 결과

에이전트 스페이스 생성

보호하려는 애플리케이션 또는 프로젝트에 대한 새 에이전트 스페이스를 생성합니다.

1. AWS Security Agent 콘솔에서 에이전트 스페이스 페이지로 이동합니다.
2. 에이전트 공간 생성을 선택합니다.
3. 에이전트 스페이스 이름 필드에 에이전트 스페이스의 이름을 입력합니다.

> **Note**
>
> 에이전트 스페이스 이름은 웹 애플리케이션의 사용자에게 표시되며이 스페이스가 나타내는 애플리케이션 또는 프로젝트를 식별하는 데 도움이 됩니다.

4. (선택 사항) 설명 필드에 에이전트 공간의 목적을 구분하는 데 도움이 되는 설명을 입력합니다.

> **Tip**
>
> 설명은 에이전트 스페이스의 목적을 구분하는 데 도움이 됩니다. "고객 포털 웹 애플리케이션", "결제 처리 마이크로서비스" 또는 "내부 분석 플랫폼"과 같이이 에이전트 스페이스가 보호할 특정 애플리케이션 또는 프로젝트를 설명하는 것이 좋습니다.

5. 생성(Create)을 선택합니다.

> **Note**
>
> AWS Security Agent가 에이전트 스페이스를 생성합니다. 이제이 애플리케이션과 관련된 기능을 구성하고 리소스를 연결할 수 있습니다.

다음 단계

에이전트 스페이스를 생성한 후:

- 코드 검토 및 침투 테스트 컨텍스트를 위해 소스 코드 리포지토리(GitHub, GitLab, Bitbucket 또는 GitHub Enterprise Server) 연결
- 설계 검토 및 침투 테스트에서 문서 컨텍스트를 위해 Confluence 연결
- 연결된 리포지토리에 대한 코드 검토 기능 활성화(참조the section called “GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화”)
- 도메인 확인을 포함한 침투 테스트 기능 구성
- (IAM Identity Center를 사용하는 경우) 에이전트 스페이스 페이지의 웹 앱 섹션에서이 에이전트 스페이스에 사용자를 할당합니다(참조the section called “사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한 부여”).
- (IAM 전용 액세스를 사용하는 경우) 콘솔 액세스 권한이 있는 사용자는이 에이전트 스페이스의 관리자 액세스 링크를 통해 웹 애플리케이션을 시작할 수 있습니다.

통합 연결

에이전트 스페이스가 코드 검토, 침투 테스트 및 위협 모델링에 사용하는 프라이빗 네트워크 연결과 함께 소스 코드 공급자(GitHub, GitLab, Bitbucket 및 GitHub Enterprise Server) 및 설명서 공급자(Confluence)를 연결하고 관리합니다.

Agent Spaces와의 통합 작동 방식

AWS Security Agent는 타사 소스 코드 및 설명서 공급자에 연결하여 에이전트가 보안 평가 중에 코드 및 설명서를 분석할 수 있도록 합니다. 특정 공급자를 연결하기 전에 통합이 에이전트 스페이스 및 기능과 어떤 관련이 있는지 이해하는 데 도움이 됩니다.

한 번 등록하고 모든 곳에서 재사용

AWS Security Agent에 공급자를 연결하려면 서로 다른 수준에서 발생하는 두 가지 단계가 필요합니다.

1. 통합(계정 수준)을 등록합니다. AWS Security Agent Management Console의 통합 페이지에서 공급자를 한 번 등록합니다. 등록은 GitHub 조직, GitLab 그룹, Bitbucket 워크스페이스 또는 Confluence 사이트와 같은 공급자 계정에 대한 승인된 연결을 나타냅니다. 등록은 계정 수준 리소스입니다.
2. 리소스를 에이전트 스페이스(에이전트 스페이스 수준)에 연결합니다. 에이전트 스페이스 내에서 소스 코드 공급자의 리포지토리 또는 Confluence의 페이지와 같은 등록된 통합의 리소스를 연결하고 에이전트가 리소스를 사용하는 방법을 구성합니다. 이 단계는 각 에이전트 스페이스에 고유합니다.

계정의 모든 에이전트 스페이스에서 단일 등록이 재사용됩니다. 한 에이전트 스페이스를 설정하는 동안 공급자를 등록한 경우 리소스를 다른 에이전트 스페이스에 연결할 때 동일한 등록을 사용할 수 있습니다. 동일한 공급자를 다시 등록하지 않습니다.

기능 간에 공유되는 하나의 연결

리소스를 에이전트 스페이스에 연결하면 해당 리소스가 에이전트의 기능 전반에서 공유됩니다. 각 기능에 대해 리포지토리를 별도로 연결하지 않습니다.

- 읽기 액세스는 리소스를 연결하여 부여됩니다. 리포지토리를 연결하면 AWS Security Agent가 전체 코드 스캔(코드 검토), 침투 테스트 컨텍스트, 위협 모델링 및 설계 검토를 위해 리포지토리를 읽을 수 있습니다. Confluence 페이지를 연결하면 에이전트가 설계 검토, 위협 모델링 및 침투 테스트의 문서 컨텍스트에 대해 읽을 수 있습니다.
- 쓰기 작업은 리포지토리별로 옵트인됩니다. 소스 코드 리포지토리를 에이전트 스페이스에 연결할 때 각 리포지토리에 대해 쓰기 작업을 추가로 활성화할 수 있습니다.
  - 코드 검토 설명 - AWS Security Agent는 검토 결과를 풀 요청(또는 GitLab의 병합 요청)에 대한 설명으로 게시합니다.
  - 코드 수정 - AWS Security Agent는 검색된 취약성에 대한 수정 사항이 포함된 풀 요청(또는 병합 요청)을 엽니다.

이러한 쓰기 작업은 퍼블릭 리포지토리에서 사용할 수 없습니다. 읽기 전용 분석은 여전히 적용됩니다.

> **Note**
>
> Confluence는 소스 코드 공급자가 아닌 설명서 공급자입니다. AWS Security Agent는 연결된 Confluence 페이지를 읽어 설계 검토, 위협 모델링 및 침투 테스트에 대한 컨텍스트를 제공합니다. 페이지를 선택하면 읽기 액세스 권한이 부여되며 페이지당 기능 토글은 없습니다.

지원되는 공급자

AWS Security Agent는 다음 공급자를 지원합니다.

- 소스 코드 - GitHub(클라우드 호스팅 GitHub 및 클라우드 호스팅 GitHub Enterprise), GitHub Enterprise Server(자체 호스팅), GitLab(클라우드 호스팅), GitLab 자체 관리형(자체 호스팅) 및 Bitbucket Cloud.
- 설명서 - Confluence Cloud.

퍼블릭 인터넷을 통해 연결할 수 없는 자체 호스팅 공급자의 경우 프라이빗 연결을 통해 에이전트의 트래픽을 라우팅할 수 있습니다. 자세한 내용은 the section called “프라이빗 호스팅 소스 제어에 연결” 단원을 참조하십시오.

다음 단계

- 통합 페이지에서 공급자를 등록합니다. 와 같은 공급자의 연결 주제를 참조하세요the section called “AWS Security Agent를 GitHub 리포지토리에 연결”.
- 리소스를 에이전트 스페이스에 연결하고 기능을 구성합니다. the section called “코드 검토 활성화” 및 the section called “침투 테스트 활성화” 단원을 참조하세요.

AWS Security Agent를 GitHub 리포지토리에 연결

AWS Security Agent를 GitHub 리포지토리에 연결하여 코드 검토, 위협 모델링, 침투 테스트 및 자동화된 문제 해결 기능을 활성화합니다. AWS Security Agent는 클라우드 호스팅 GitHub와 클라우드 호스팅 GitHub Enterprise를 모두 지원합니다. 시작하기 전에 the section called “Agent Spaces와의 통합 작

동 방식"를 검토하여 등록이 에이전트 스페이스에서 재사용되고 기능 간에 공유되는 방법을 이해합니다.

GitHub 통합은 다음과 같은 여러 용도로 사용됩니다.

> **Note**
>
> 이 페이지에서는 클라우드 호스팅 GitHub(github.com) 및 클라우드 호스팅 GitHub Enterprise를 다룹니다. 자체 호스팅 GitHub Enterprise Server는 섹션을 참조하세요the section called "AWS Security Agent를 GitHub Enterprise Server에 연결".

- 코드 검토 - 조직 보안 요구 사항에 따라 각 풀 요청의 코드 변경 사항을 자동으로 분석하고 온디맨드 전체 리포지토리 스캔을 실행합니다.
- 위협 모델링 - 소스 코드, 데이터 흐름 및 아키텍처를 분석하여 애플리케이션에 대한 이해 제공
- 침투 테스트 컨텍스트 - 소스 코드를 분석하여 침투 테스트에 대한 애플리케이션 이해 제공
- 자동 문제 해결 - 보안 평가 중에 발견된 취약성에 대한 수정 사항이 포함된 풀 요청 제출

GitHub를 AWS Security Agent에 연결하려면 GitHub 조직 또는 사용자 계정에 대한 AWS Security Agent GitHub 앱을 승인한 다음 AWS 콘솔에 연결을 등록해야 합니다.

GitHub 통합 작동 방식

풀 요청 분석은 GitHub 내에서 수행됩니다. GitHub 앱에 권한을 부여하고, 리포지토리를 연결하고, AWS Management Console에서 코드 검토 주석을 활성화하면 AWS Security Agent는 자동으로 각 새 풀 요청의 변경 사항(변경된 코드만 차등 스캔)을 스캔하고 결과를 풀 요청 주석으로 게시합니다.

GitHub가 아닌 AWS Security Agent 웹 애플리케이션에서 리포지토리의 전체 코드베이스를 스캔하는 전체 코드 검토를 생성하고 실행합니다.

침투 테스트 및 위협 모델링은 AWS Security Agent 웹 애플리케이션 내에서 시작됩니다. 사용자는 연결된 리포지토리를 선택하여 애플리케이션 컨텍스트를 제공하고 침투 테스트를 위한 대상 도메인을 지정합니다. 자동 문제 해결을 활성화하면 사용자는 연결된 리포지토리에 대한 풀 요청을 열어 AWS Security Agent에 조사 결과를 수정하도록 요청할 수 있습니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- GitHub 조직 관리자 액세스 또는 GitHub 사용자 계정 소유자 액세스
- AWS 관리 콘솔에서 에이전트 스페이스에 대한 통합을 구성할 수 있는 권한
- 코드 검토, 위협 모델링 및 침투 테스트를 위해 연결할 리포지토리 이해

Important

GitHub 앱은 GitHub 계정 또는 GitHub 조직에 한 번만 설치할 수 있습니다. 동일한 GitHub 조직을 AWS Security Agent에 연결해야 하는 경우 통합이 처음 등록된 동일한 AWS 계정을 사용해야 합니다.

Note

GitHub 엔터프라이즈 조직에서 IP 허용 목록을 활성화한 경우 GitHub 앱에서 허용되는 IP 주소를 수락해야 합니다. 허용 목록에 IP 주소를 자동으로 추가하도록 선택할 수도 있습니다. 자세한 내용은 GitHub 설명서의 GitHub 앱에 의한 액세스 허용 및 허용된 IP 주소 활성화를 참조하세요. GitHub
다음 IP 주소는 GitHub 리소스에 액세스하는 데 사용됩니다.

- 미국 동부(버지니아 북부)(us-east-1)
  - `34.228.181.128`
  - `44.219.176.187`
  - `54.226.244.221`
- 미국 서부(오레곤)(us-west-2)
  - `34.212.16.133`
  - `52.89.67.212`
  - `54.187.135.61`
- 아시아 태평양(뭄바이)(ap-south-1)
  - `13.126.209.199`
  - `13.234.6.24`
  - `35.154.102.216`
- 아시아 태평양(싱가포르)(ap-southeast-1)
  - `18.139.13.125`

    - 47.130.240.215
    - 54.179.238.173
  - 아시아 태평양(시드니)(ap-southeast-2)
    - 13.237.95.197
    - 13.238.84.102
    - 52.64.174.242
  - 아시아 태평양(도쿄)(ap-northeast-1)
    - 13.192.12.233
    - 35.74.181.230
    - 57.183.50.158
  - 유럽(프랑크푸르트)(eu-central-1)
    - 18.158.110.140
    - 52.57.96.160
    - 52.59.55.56
  - 유럽(아일랜드)(eu-west-1)
    - 34.251.85.24
    - 52.30.157.157
    - 52.51.192.222
  - 남아메리카(상파울루)(sa-east-1)
    - 54.94.247.213
    - 54.207.222.14
    - 54.232.201.242

AWS Security Agent GitHub 앱 권한 부여 및 등록

AWS Security Agent GitHub 앱이 GitHub 조직 또는 사용자 계정에 액세스하도록 권한을 부여한 다음 AWS 콘솔에 연결을 등록합니다.

> **Important**
>
> 브라우저를 닫거나 탐색하지 않고이 프로세스의 모든 단계를 완료합니다. 등록 프로세스가 중단되면 GitHub 앱을 제거하고 다시 시작해야 할 수 있습니다.

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 추가(Add Integrations)를 선택합니다.
3. GitHub를 선택합니다.
4. 다음을 선택합니다.
5. 설치 및 권한 부여를 선택합니다.

   권한 부여를 완료하기 위해 GitHub로 리디렉션됩니다. 연결하려는 조직 또는 사용자 계정에 대한 관리자 액세스 권한이 있는 계정으로 GitHub에 로그인했는지 확인합니다.
6. GitHub에서 AWS Security Agent GitHub 앱을 설치할 계정 또는 조직을 선택합니다.
7. AWS Security Agent가 액세스할 수 있는 리포지토리를 선택합니다.
   - 모든 리포지토리 - 조직 또는 사용자 계정의 모든 현재 및 미래 리포지토리에 대한 액세스 권한 부여
   - 리포지토리만 선택 - 드롭다운에서 특정 리포지토리를 선택합니다. 한 번에 여러 리포지토리를 선택할 수 있습니다.

   > **Note**
   >
   > GitHub 조직 또는 사용자 계정 설정의 GitHub 앱 설정을 방문하여 언제든지 리포지토리 액세스를 수정할 수 있습니다.
8. 설치 및 권한 부여를 선택합니다.
9. 등록을 완료하기 위해 AWS Management Console로 다시 리디렉션됩니다.
10. 등록 세부 정보 섹션에서 다음 필드를 구성합니다.
    a. 등록 이름 -이 GitHub 연결에 대한 설명 이름을 입력합니다. "Acme-Corp-Org" 또는 "Production-Repos"와 같이 GitHub 조직 또는 사용자 계정을 식별하는 이름을 사용합니다.
    b. 계정 유형 - 드롭다운에서 다음 중 하나를 선택합니다.
       - 조직 - GitHub 조직 계정을 연결한 경우
       - 사용자 - 개인 GitHub 사용자 계정을 연결한 경우

c. 조직 이름(조직을 선택한 경우에만 표시됨) - GitHub에 표시되는 GitHub 조직의 정확한 이름을 입력합니다.

11.연결을 선택합니다.

12.확인 메시지가 표시되고 통합 페이지로 돌아가면 새 GitHub 연결이 등록 이름과 함께 나타납니다. 추가 GitHub 조직 또는 사용자 계정을 연결하려면 통합 추가를 다시 선택하여이 프로세스를 반복합니다.

GitHub 통합 문제 해결

GitHub 통합 프로세스 중에 문제가 발생하는 경우 다음 지침을 사용하여 일반적인 문제를 해결합니다.

등록을 완료할 수 없음

등록 프로세스를 완료할 수 없는 경우(예: 브라우저를 닫았거나, 등록 페이지에서 다른 곳으로 이동했거나, 세션이 중단된 경우) AWS Security Agent GitHub 앱이 GitHub 조직에 설치되지만 AWS 콘솔에는 등록되지 않을 수 있습니다.

증상:

- GitHub 앱을 다시 승인하려고 하면 GitHub에 "설치" 대신 "구성"이 표시됩니다.
- AWS 콘솔에서 등록을 완료할 수 없습니다.
- 통합이 통합 목록에 표시되지 않습니다.

해결 방법:

1. GitHub 조직 또는 사용자 계정에서 AWS Security Agent GitHub 앱을 제거합니다.
2. AWS Security Agent 콘솔로 돌아가서 처음부터 통합 프로세스를 다시 시작합니다.

동일한 GitHub 조직을 통합하려는 여러 AWS 계정

GitHub 앱은 GitHub 계정 또는 GitHub 조직에 한 번만 설치할 수 있습니다. 다른 AWS Security Agent 계정과 이미 통합된 GitHub 조직의 리포지토리를 사용해야 하는 경우 통합이 처음 등록된 AWS 계정을 사용해야 합니다.

해결 방법:

- GitHub 통합이 등록된 AWS Security Agent 계정 식별

- 해당 AWS 계정을 사용하여 에이전트 스페이스를 생성하고 리포지토리를 연결합니다.
- 통합을 다른 AWS 계정으로 이동해야 하는 경우 먼저 원래 AWS 계정에서 GitHub 앱을 제거한 다음 새 계정과 통합합니다.

다음 단계

GitHub를 AWS Security Agent에 연결한 후:

- 이러한 리포지토리를 사용할 에이전트 공간으로 이동합니다.
- 코드 검토 활성화 또는 침투 테스트 설정을 선택하여 특정 리포지토리를 에이전트 스페이스에 연결하고 사용량을 구성합니다.
- AWS Security Agent가 취약성 수정으로 풀 요청을 제출할 수 있도록 자동 수정 활성화
- GitHub 조직 설정에서 GitHub 앱 권한 및 리포지토리 액세스 검토

GitHub 통합 제거

AWS Security Agent가 특정 GitHub 조직 또는 사용자 계정의 리포지토리에 더 이상 액세스할 필요가 없는 경우 GitHub 통합을 제거합니다. AWS 콘솔에서 통합을 제거하기 전에 먼저 GitHub에서 AWS Security Agent GitHub 앱을 제거해야 합니다.

제거를 위한 사전 조건

GitHub 통합을 제거하기 전에 다음이 있는지 확인합니다.

- 이 통합에서 연결된 리포지토리가 있는 에이전트 스페이스 확인
- 영향 이해:이 통합을 제거하면이 통합의 리포지토리를 사용하여 모든 에이전트 스페이스에서 코드 검토, 침투 테스트 컨텍스트 및 침투 테스트 문제 해결을 위한 리포지토리 연결이 끊어집니다.
- GitHub 조직 관리자 액세스 또는 GitHub 앱을 제거하기 위한 사용자 계정 소유자 액세스

1단계: GitHub에서 AWS 보안 에이전트 GitHub 앱 제거

먼저 GitHub 조직 또는 사용자 계정에서 AWS Security Agent GitHub 앱을 제거합니다.

GitHub Organizations의 경우:

1. github.com 이동하여 조직 페이지를 엽니다.
2. 왼쪽 사이드바에서 설정을 선택합니다.

3. 코드, 계획 및 자동화에서 설치된 GitHub 앱을 선택합니다.
4. 목록에서 AWS Security Agent 앱을 찾습니다.
5. AWS Security Agent 앱 옆의 구성을 선택합니다.
6. 구성 페이지 하단으로 스크롤하여 제거를 선택합니다.
7. 메시지가 표시되면 제거를 확인합니다.

GitHub 사용자 계정의 경우:

1. github.com 이동합니다.
2. 프로필 사진을 선택합니다.
3. 설정을 선택합니다.
4. 왼쪽 사이드바에서 애플리케이션을 선택합니다.
5. 설치된 GitHub 앱 탭을 엽니다.
6. 목록에서 AWS Security Agent 앱을 찾습니다.
7. AWS Security Agent 앱 옆에 있는 구성을 선택합니다.
8. 구성 페이지 하단으로 스크롤하여 제거를 선택합니다.
9. 메시지가 표시되면 제거를 확인합니다.

2단계: AWS Security Agent에서 통합 제거

GitHub 앱을 제거한 후 AWS 콘솔에서 통합 등록을 제거합니다.

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 목록에서 제거하려는 GitHub 통합을 찾습니다.
3. 통합을 클릭하여 선택합니다.
4. 제거를 선택합니다.
5. 영향을 경고하는 확인 대화 상자를 검토합니다.

> **Warning**
>
> 이 통합을 제거하면이 GitHub 조직 또는 사용자 계정에서 연결된 리포지토리가 있는 모든 에이전트 스페이스에 영향을 미칩니다. 이렇게 하면 다음과 같은 문제가 발생합니다.
>
> - 연결된 리포지토리에 대한 코드 검토 기능
> - 연결된 리포지토리의 침투 테스트 컨텍스트

- 연결된 리포지토리의 침투 테스트 문제 해결 기능

계속하기 전에 GitHub에서 AWS 보안 에이전트 GitHub 앱을 제거했는지 확인합니다.

6. 아직 GitHub에서 GitHub 앱을 제거하지 않은 경우 경고가 표시됩니다. 먼저 1단계로 돌아가서 제거를 완료합니다.
7. GitHub 앱을 제거하고 영향을 이해한 경우 제거 확인을 선택합니다.
8. 통합 목록에서 통합이 제거됩니다. 이 통합의 리포지토리가 있는 모든 에이전트 스페이스는 더 이상 코드 검토, 침투 테스트 컨텍스트 또는 자동 문제 해결을 위해 해당 리포지토리에 액세스할 수 없습니다.

AWS Security Agent를 GitLab 리포지토리에 연결

AWS Security Agent를 GitLab Cloud 리포지토리에 연결하여 코드 검토, 위협 모델링, 침투 테스트 및 자동화된 문제 해결 기능을 활성화합니다. 시작하기 전에 the section called “Agent Spaces와의 통합 작동 방식”를 검토하여 등록이 에이전트 스페이스에서 재사용되고 기능 간에 공유되는 방법을 이해합니다.

GitLab 통합은 여러 용도로 사용됩니다.

- 코드 검토 - 조직 보안 요구 사항에 따라 각 병합 요청의 코드 변경 사항을 자동으로 분석하고 온디맨드 전체 리포지토리 스캔을 실행합니다.
- 위협 모델링 - 소스 코드, 데이터 흐름 및 아키텍처를 분석하여 애플리케이션에 대한 이해 제공
- 침투 테스트 컨텍스트 - 소스 코드를 분석하여 침투 테스트에 대한 애플리케이션 이해 제공
- 자동 문제 해결 - 보안 평가 중에 발견된 취약성에 대한 수정 사항이 포함된 병합 요청 제출

GitLab을 AWS Security Agent에 연결하려면 적절한 권한이 있는 GitLab 액세스 토큰을 제공한 다음 AWS 콘솔에 연결을 등록해야 합니다.

GitLab 통합 작동 방식

병합 요청 분석은 GitLab 내에서 수행됩니다. AWS Management Console에서 액세스 토큰을 제공하고, 프로젝트를 연결하고, 코드 검토 설명을 활성화하면 AWS Security Agent는 각 새 병합 요청의 변경 사항(변경된 코드만 차등 스캔)을 자동으로 스캔하고 결과를 병합 요청 설명으로 게시합니다.

GitLab이 아닌 AWS Security Agent 웹 애플리케이션에서 프로젝트의 전체 코드베이스를 스캔하는 전체 코드 검토를 생성하고 실행합니다.

침투 테스트 및 위협 모델링은 AWS Security Agent 웹 애플리케이션 내에서 시작됩니다. 사용자는 대상 도메인을 지정하고 연결된 리포지토리를 선택하여 애플리케이션 컨텍스트를 제공합니다. 자동 수정을 활성화하면 사용자는 연결된 리포지토리에 대한 병합 요청을 열어 AWS Security Agent에 조사 결과 수정을 요청할 수 있습니다.

> **Note**
>
> 취약성이 수정되기 전에 공개되지 않도록 퍼블릭 GitLab 리포지토리에서는 자동 문제 해결을 사용할 수 없습니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 연결하려는 프로젝트에 대한 관리자 또는 소유자 액세스 권한이 있는 GitLab.com 계정
- 연결 유형에 필요한 범위가 있는 GitLab 액세스 토큰:
  - 개인 - 모든 읽기 권한과 `api` 권한이 있는 개인 액세스 토큰입니다.
  - 그룹 - `read_api` 및 `read_repository` 범위가 있는 그룹 액세스 토큰입니다.
- AWS Security Agent Management Console에서 통합을 구성할 수 있는 권한

> **Important**
>
> 토큰 만료를 현재 날짜 이후 최대 365일로 설정합니다.

> **Note**
>
> GitLab 개인용 액세스 토큰은 여러 AWS 계정에서 사용할 수 있습니다. GitHub 및 Atlassian 통합과 달리 동일한 GitLab 계정을 여러 AWS Security Agent 인스턴스에 연결하는 데는 제한이 없습니다.

GitLab 연결 등록

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 추가(Add Integrations)를 선택합니다.

3. GitLab을 선택한 후 다음을 선택합니다.
4. 계정 유형 선택에서 다음 중 하나를 선택합니다.
   - 개인 - 개별 GitLab 사용자 계정을 연결합니다.
   - 그룹 - 여러 프로젝트가 포함된 GitLab 그룹을 연결합니다. 그룹 ID를 입력합니다.
5. 액세스 토큰 필드에 GitLab 액세스 토큰을 붙여 넣습니다.
6. 등록 이름 필드에와 같이이 연결을 설명하는 이름을 입력합니다`Engineering-Team-GitLab`. 유효한 문자는 문자, 숫자, 마침표, 밑줄 및 하이픈입니다.
7. 연결을 선택합니다.

   통합 페이지로 돌아가면 등록 이름과 함께 새 연결이 나타납니다.

GitLab 통합 문제 해결

GitLab 통합 프로세스 중에 문제가 발생하면 다음 지침을 사용하여 일반적인 문제를 해결하세요.

유효하지 않거나 만료된 토큰

증상

- 통합 연결 실패
- 이전에 작동하던 통합 작동 중지
- 인증 실패를 나타내는 오류 메시지

해결 방법

1. GitLab에서 사용자 설정으로 이동하여 토큰 액세스를 선택합니다.
2. 토큰이 만료되지 않았는지 확인합니다.
3. 토큰의 유형에 필요한 범위가 있는지 확인합니다.
4. 토큰이 만료된 경우 새 토큰을 생성하고 AWS 콘솔에서 통합을 업데이트합니다.

속도 제한

증상

- 코드 검토 중 간헐적 실패

- 지연된 병합 요청 분석

해결 방법

- GitLab은 API 요청에 속도 제한을 적용합니다. 리포지토리 수가 많거나 병합 요청이 자주 발생하는 경우 일부 요청이 제한될 수 있습니다.
- 속도 제한 기간이 재설정될 때까지 기다리거나 GitLab 지원팀에 문의하여 속도 제한을 늘리세요.

다음 단계

GitLab을 AWS Security Agent에 연결한 후:

- 이러한 리포지토리를 사용할 에이전트 공간으로 이동합니다.
- 코드 검토 활성화 또는 침투 테스트 설정을 선택하여 특정 프로젝트를 에이전트 스페이스에 연결하고 사용량을 구성합니다.
- AWS Security Agent가 취약성 수정과 함께 병합 요청을 제출할 수 있도록 코드 수정 활성화
- 프라이빗 호스팅 GitLab 인스턴스는 섹션을 참조하세요. the section called “AWS Security Agent를 GitLab 자체 관리형에 연결”

AWS Security Agent를 GitLab 자체 관리형에 연결

AWS Security Agent를 GitLab 자체 관리형 인스턴스에 연결하여 자체 인프라에서 호스팅되는 리포지토리에 대한 코드 검토, 위협 모델링, 침투 테스트 및 자동화된 문제 해결 기능을 활성화합니다.

GitLab 자체 관리형 통합은 프라이빗 인스턴스에 대한 네트워크 연결을 위한 추가 구성과 함께 GitLab 클라우드(참조the section called “AWS Security Agent를 GitLab 리포지토리에 연결”)와 동일하게 작동합니다. 시작하기 전에 the section called “Agent Spaces와의 통합 작동 방식”를 검토하여 등록이 에이전트 스페이스 간에 재사용되고 기능 간에 공유되는 방법을 이해합니다.

> **Note**
>
> GitLab 자체 관리형은 별도의 통합 유형이 아니라 GitLab 통합을 통해 등록됩니다. 등록 흐름에서 GitLab 자체 호스팅 엔드포인트 사용을 선택하고 인스턴스 URL을 제공합니다. 클라우드 호스팅 GitLab 흐름은에 설명되어 있습니다the section called “AWS Security Agent를 GitLab 리포지토리에 연결”.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 다음 중 하나인 GitLab 자체 관리형 인스턴스:
  - 인터넷을 통해 공개적으로 액세스할 수 있습니다. 또는
  - 프라이빗 연결을 통해 액세스 가능(참조the section called “프라이빗 호스팅 소스 제어에 연결”)
- 연결 유형에 필요한 범위가 있는 GitLab 액세스 토큰:
  - 개인 - 모든 읽기 권한과 `api` 권한이 있는 개인 액세스 토큰입니다.
  - 그룹 - `read_api` 및 `read_repository` 범위가 있는 그룹 액세스 토큰입니다.
- 연결하려는 프로젝트에 대한 관리자 또는 소유자 액세스
- GitLab 인스턴스는 최소 TLS 버전이 1.2인 HTTPS 트래픽을 제공해야 합니다.

> **Note**
>
> GitLab 자체 관리형 인스턴스가 프라이빗 인증 기관에서 발급한 TLS 인증서를 사용하는 경우 프라이빗 연결을 생성할 때 인증서의 PEM 인코딩 퍼블릭 키를 제공할 수 있습니다. 이렇게 하면 AWS Security Agent가 인스턴스에 대한 TLS 연결을 신뢰할 수 있습니다.

GitLab 자체 관리형 연결 등록

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 추가(Add Integrations)를 선택합니다.
3. GitLab을 선택한 후 다음을 선택합니다.
4. 계정 유형 선택에서 개인 또는 그룹을 선택합니다. 그룹을 선택한 경우 그룹 ID를 입력합니다.
5. GitLab 자체 호스팅 엔드포인트 사용을 선택합니다.
6. GitLab 자체 호스팅 엔드포인트 URL 필드에와 같이 인스턴스의 URL을 입력합니다`https://gitlab.example.com`.
7. 인스턴스에 공개적으로 액세스할 수 없는 경우 프라이빗 연결을 사용하여 엔드포인트에 연결을 선택한 다음 기존 프라이빗 연결을 선택하거나 새 프라이빗 연결을 생성합니다. the section called “프라이빗 호스팅 소스 제어에 연결”을(를) 참조하세요.
8. 액세스 토큰 필드에 GitLab 액세스 토큰을 붙여 넣습니다.

9. 등록 이름 필드에이 연결을 설명하는 이름을 입력합니다. 유효한 문자는 문자, 숫자, 마침표, 밑줄 및 하이픈입니다.
10. 연결을 선택합니다.

    통합 페이지로 돌아가면 새 연결이 등록 이름과 함께 나타납니다.

프라이빗 연결

GitLab 자체 관리형 인스턴스에 공개적으로 액세스할 수 없는 경우 통합을 등록하기 전에 프라이빗 연결을 생성해야 합니다. 자세한 지침은 the section called “프라이빗 호스팅 소스 제어에 연결” 단원을 참조하십시오.

> **Important**
>
> 서비스 관리형 프라이빗 연결을 사용하려면 에이전트 스페이스가 생성된 동일한 AWS 계정에서 GitLab 자체 관리형 인스턴스를 실행해야 합니다. 교차 계정 액세스의 경우 자체 VPC Lattice 리소스 구성을 제공하는 자체 관리형 프라이빗 연결을 사용합니다.

GitLab 자체 관리형 통합 문제 해결

의 문제 해결 단계 외에도 다음 the section called “AWS Security Agent를 GitLab 리포지토리에 연결”문제는 자체 관리형 인스턴스에만 해당됩니다.

인스턴스에 연결할 수 없음

증상

- 제한 시간 또는 네트워크 오류로 인해 연결이 실패함
- 통합이 이전에 작동했지만 작동을 중지함

해결 방법

- GitLab 인스턴스가 실행 중이고 액세스 가능한지 확인
- 프라이빗 연결을 사용하는 경우 VPC Lattice 리소스 게이트웨이가 정상이고 ENIs가 인스턴스에 네트워크로 연결되어 있는지 확인합니다.
- 보안 그룹이 구성된 포트에서 트래픽을 허용하는지 확인
- TLS 인증서가 유효하고 만료되지 않았는지 확인

TLS 인증서 오류

증상

- SSL/TLS 오류와 함께 연결 실패

해결 방법

- 인스턴스가 TLS 1.2 이상에서 HTTPS를 제공하는지 확인
- 프라이빗 인증 기관을 사용하는 경우 프라이빗 연결 설정 중에 PEM 인코딩 퍼블릭 키가 제공되었는지 확인합니다.
- 인증서가 만료되지 않았는지 확인

다음 단계

GitLab 자체 관리형를 AWS 보안 에이전트에 연결한 후:

- 이러한 리포지토리를 사용할 에이전트 공간으로 이동합니다.
- 코드 검토 활성화 또는 침투 테스트 설정을 선택하여 특정 프로젝트를 연결합니다.
- 병합 요청 기반 수정 사항에 대한 코드 수정 활성화

GitLab 통합 제거

AWS Security Agent가 특정 GitLab 계정 또는 그룹의 리포지토리에 더 이상 액세스할 필요가 없는 경우 GitLab 통합을 제거합니다.

제거를 위한 사전 조건

GitLab 통합을 제거하기 전에 다음이 있는지 확인합니다.

- 이 통합에서 연결된 리포지토리가 있는 에이전트 스페이스 확인
- 영향 이해:이 통합을 제거하면이 통합의 리포지토리를 사용하여 모든 에이전트 스페이스에서 코드 검토, 침투 테스트 컨텍스트, 위협 모델링 및 자동 문제 해결을 위한 리포지토리 연결이 끊어집니다.

AWS Security Agent에서 통합 제거

1. AWS Security Agent Management Console에서 통합으로 이동합니다.

2. 제거할 GitLab 통합을 찾습니다.
3. 통합을 클릭하여 선택합니다.
4. 제거를 선택합니다.
5. 확인 대화 상자를 검토하고 제거 확인을 선택합니다.

> **Note**
>
> 통합을 제거한 후 추가 액세스를 방지하기 위해 GitLab에서 개인 액세스 토큰을 취소할 수도 있습니다. GitLab 사용자 설정으로 이동하여 토큰을 삭제하거나 취소합니다.

GitLab 자체 관리형 통합에도 동일한 프로세스가 적용됩니다.

AWS Security Agent를 GitHub Enterprise Server에 연결

AWS Security Agent를 GitHub Enterprise Server(GHES) 인스턴스에 연결하여 자체 인프라에서 호스팅되는 리포지토리에 대한 코드 검토, 위협 모델링, 침투 테스트 및 자동 문제 해결 기능을 활성화합니다.

GitHub Enterprise Server 통합은 자체 호스팅 인스턴스에 대한 네트워크 연결을 위한 추가 구성과 함께 클라우드 호스팅 GitHub(GitHub 리포지토리에 AWS 보안 에이전트 연결 참조). 시작하기 전에 the section called “Agent Spaces와의 통합 작동 방식”를 검토하여 등록이 에이전트 스페이스에서 재사용되고 기능 간에 공유되는 방법을 이해합니다.

> **Note**
>
> GitHub Enterprise Server는 별도의 통합 유형이 아니라 GitHub 통합을 통해 등록됩니다. 등록 흐름에서 GitHub Enterprise Server를 인스턴스 유형으로 선택합니다. 클라우드 호스팅 GitHub.com 흐름은에 설명되어 있습니다the section called “AWS Security Agent를 GitHub 리포지토리에 연결”.

GitHub Enterprise Server 통합 작동 방식

풀 요청 분석은 GitHub Enterprise Server 내에서 수행됩니다. AWS Management Console에서 연결을 승인하고, 리포지토리를 연결하고, 코드 검토 설명을 활성화하면 AWS Security Agent는 각 새 풀 요청의 변경 사항(변경된 코드만 차등 스캔)을 자동으로 스캔하고 결과를 풀 요청 설명으로 게시합니다.

GitHub Enterprise Server가 아닌 AWS Security Agent 웹 애플리케이션에서 리포지토리의 전체 코드 베이스를 스캔하는 전체 코드 검토를 생성하고 실행합니다.

침투 테스트 및 위협 모델링은 AWS Security Agent 웹 애플리케이션 내에서 시작됩니다. 사용자는 대상 도메인을 지정하고 연결된 리포지토리를 선택하여 애플리케이션 컨텍스트를 제공합니다. 자동 문제 해결을 활성화하면 사용자는 연결된 리포지토리에 대한 풀 요청을 열어 AWS Security Agent에 조사 결과를 수정하도록 요청할 수 있습니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 다음 중 하나인 GitHub Enterprise Server 인스턴스:
  - 인터넷을 통해 공개적으로 액세스 가능 또는
  - 프라이빗 연결을 통해 액세스 가능(참조the section called “프라이빗 호스팅 소스 제어에 연결”)
- GHES 인스턴스에서 사이트 관리자 또는 조직 관리자 액세스
- GHES 인스턴스는 최소 TLS 버전이 1.2인 HTTPS 트래픽을 제공해야 합니다.
- AWS Security Agent Management Console에서 통합을 구성할 수 있는 권한

> **Note**
>
> GitHub Enterprise Server 통합은 여러 AWS 계정에서 사용할 수 있습니다.

GitHub Enterprise Server 연결 등록

GitHub Enterprise Server 연결을 등록하려면 OAuth 기반 권한 부여 흐름을 사용합니다.

> **Important**
>
> 브라우저를 닫거나 탐색하지 않고이 프로세스의 모든 단계를 완료합니다. 등록 프로세스가 중단되면 처음부터 다시 시작해야 할 수 있습니다.

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 추가(Add Integrations)를 선택합니다.

3. GitHub를 선택한 후 다음을 선택합니다.
4. 인스턴스 유형에서 GitHub Enterprise Server를 선택합니다.
5. GitHub Enterprise Server URL 필드에와 같이 인스턴스의 HTTPS URL을 입력합니다`https://github.example.com`.
6. 인스턴스에 공개적으로 액세스할 수 없는 경우 프라이빗 연결을 사용하여 엔드포인트에 연결을 선택한 다음 기존 프라이빗 연결을 선택하거나 새 프라이빗 연결을 생성합니다. the section called “프라이빗 호스팅 소스 제어에 연결”을(를) 참조하세요.
7. 세부 정보 등록 섹션에서 다음 필드를 구성합니다.
   a. 등록 이름 -이 연결을 설명하는 이름을 입력합니다. 유효한 문자는 문자, 숫자, 마침표, 밑줄 및 하이픈입니다.
   b. GitHub 계정 유형 - 조직 또는 사용자를 선택합니다.
   c. 조직 이름(조직을 선택한 경우에만 표시됨) - GitHub Enterprise Server 조직의 정확한 이름을 입력합니다. 이름은 대/소문자를 구분합니다.
8. 연결을 선택합니다.

> **Note**
>
> AWS Security Agent는 콘솔에서 리디렉션하여 GitHub Enterprise Server 인스턴스에 대한 권한 부여를 완료합니다. 권한 부여가 완료되면 콘솔로 돌아가서 통합 페이지에 새 연결이 나타납니다.

프라이빗 연결

GitHub Enterprise Server 인스턴스에 공개적으로 액세스할 수 없는 경우 통합을 등록하기 전에 프라이빗 연결을 생성해야 합니다. 자세한 지침은 the section called “프라이빗 호스팅 소스 제어에 연결” 단원을 참조하십시오.

> **Important**
>
> 서비스 관리형 프라이빗 연결을 사용하려면 에이전트 스페이스가 생성된 동일한 AWS 계정에서 GHES 인스턴스를 실행해야 합니다. 교차 계정 액세스의 경우 자체 관리형 프라이빗 연결을 사용합니다.

> **Note**
>
> GHES 인스턴스가 프라이빗 인증 기관에서 발급한 TLS 인증서를 사용하는 경우 프라이빗 연결을 생성할 때 인증서의 PEM 인코딩 퍼블릭 키를 제공합니다.

GitHub Enterprise Server 통합 문제 해결

AWS Security Agent를 GitHub Enterprise Server에 연결하는 데 문제가 발생하면 다음 지침을 사용하여 일반적인 문제를 진단하고 해결합니다.

OAuth 리디렉션 실패

증상

- 권한 부여 흐름 중에 브라우저 리디렉션 실패
- GHES에서 권한을 부여한 후 표시되는 오류 페이지

해결 방법

- 브라우저에서 GHES 인스턴스에 액세스할 수 있는지 확인
- OAuth 콜백 URL이 올바르게 구성되었는지 확인
- 처음부터 통합 프로세스를 다시 시작합니다.

인스턴스에 연결할 수 없음

증상

- 제한 시간 또는 네트워크 오류로 인해 연결이 실패함

해결 방법

- GHES 인스턴스가 실행 중이고 액세스 가능한지 확인
- 프라이빗 연결을 사용하는 경우 VPC Lattice 연결을 확인합니다.
- 보안 그룹이 구성된 포트에서 트래픽을 허용하는지 확인
- TLS 인증서가 유효한지 확인(TLS 최소 1.2)

다음 단계

GitHub Enterprise Server를 AWS Security Agent에 연결한 후:

- 이러한 리포지토리를 사용할 에이전트 공간으로 이동합니다.
- 코드 검토 활성화 또는 침투 테스트 설정을 선택하여 특정 리포지토리 연결
- AWS Security Agent가 취약성 수정과 함께 풀 요청을 제출할 수 있도록 코드 수정 활성화

GitHub Enterprise Server 통합 제거

AWS Security Agent가 특정 GHES 인스턴스의 리포지토리에 더 이상 액세스할 필요가 없는 경우 GitHub Enterprise Server 통합을 제거합니다.

제거를 위한 사전 조건

GHES 통합을 제거하기 전에:

- 이 통합에서 연결된 리포지토리가 있는 에이전트 스페이스 확인
- 영향 이해:를 제거하면 연결된 모든 리포지토리에 대한 코드 검토, 침투 테스트 컨텍스트, 위협 모델링 및 자동 문제 해결이 중단됩니다.

통합 제거

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 제거하려는 GitHub Enterprise Server 통합을 찾습니다.
3. 통합을 선택합니다.
4. 제거를 선택합니다.
5. 확인 대화 상자를 검토하고 제거 확인을 선택합니다.

> **Note**
>
> 제거 후 GHES 인스턴스에서 OAuth 애플리케이션을 취소할 수도 있습니다. GHES 조직 설정으로 이동하여 승인된 애플리케이션을 제거합니다.

Atlassian 설치 ID 찾기

AWS 콘솔에 Bitbucket 또는 Confluence 통합을 등록하기 전에 Atlassian 사이트에 설치된 AWS Security Agent Forge 앱에서 설치 ID를 찾아야 합니다. 이 ID를 등록 흐름에 붙여 넣습니다.

Bitbucket의 경우

1. Bitbucket에서 Workspace 설정으로 이동합니다.
2. 사이드바에서 앱 위조를 선택합니다.
3. 설치된 앱 목록에서 AWS Security Agent와 연결을 찾습니다.
4. 클립보드에 복사를 선택하여 설치 ID를 복사합니다.

Confluence의 경우

1. Confluence에서 Confluence 관리로 이동합니다.
2. 사이드바에서 앱을 선택합니다.
3. 설치된 앱 목록에서 AWS Security Agent와 연결을 찾습니다.
4. 클립보드에 복사를 선택하여 설치 ID를 복사합니다.

AWS Security Agent Management Console에서 등록을 완료할 때이 설치 ID가 필요합니다.

AWS Security Agent를 Bitbucket 리포지토리에 연결

AWS Security Agent를 Bitbucket Cloud 리포지토리에 연결하여 코드 검토, 위협 모델링, 침투 테스트 및 자동화된 문제 해결 기능을 활성화합니다. 시작하기 전에 the section called “Agent Spaces와의 통합 작동 방식”를 검토하여 등록이 에이전트 스페이스에서 재사용되고 기능 간에 공유되는 방법을 이해합니다.

Bitbucket 통합은 여러 가지 용도로 사용됩니다.

- 코드 검토 - 조직 보안 요구 사항에 따라 각 풀 요청의 코드 변경 사항을 자동으로 분석하고 온디맨드 전체 리포지토리 스캔을 실행합니다.
- 위협 모델링 - 소스 코드, 데이터 흐름 및 아키텍처를 분석하여 애플리케이션에 대한 이해 제공
- 침투 테스트 컨텍스트 - 침투 테스트에 대한 애플리케이션 이해 제공
- 자동 문제 해결 - 보안 평가 중에 발견된 취약성에 대한 수정 사항이 포함된 풀 요청 제출

Bitbucket을 AWS Security Agent에 연결하려면 Atlassian 사이트에 AWS Security Agent Forge 앱을 설치하고 OAuth 권한 부여 흐름을 완료해야 합니다.

> **Note**
>
> AWS Security Agent는 Bitbucket Cloud만 지원합니다. Bitbucket Server 및 Bitbucket Data Center는 지원되지 않습니다.

Bitbucket 통합 작동 방식

풀 요청 분석은 Bitbucket 내에서 수행됩니다. Forge 앱을 설치하고, 리포지토리를 연결하고, AWS Management Console에서 코드 검토 설명을 활성화하면 AWS Security Agent는 각 새 풀 요청의 변경 사항(변경된 코드만 차등 스캔)을 자동으로 스캔하고 결과를 풀 요청 설명으로 게시합니다.

Bitbucket이 아닌 AWS Security Agent 웹 애플리케이션에서 리포지토리의 전체 코드베이스를 스캔하는 전체 코드 검토를 생성하고 실행합니다.

침투 테스트 및 위협 모델링은 AWS Security Agent 웹 애플리케이션 내에서 시작됩니다. 사용자는 대상 도메인을 지정하고 연결된 리포지토리를 선택하여 애플리케이션 컨텍스트를 제공합니다.

> **Note**
>
> 취약성이 수정되기 전에 공개되지 않도록 퍼블릭 Bitbucket 리포지토리에서는 자동 수정을 사용할 수 없습니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 관리자 액세스 권한이 있는 Bitbucket Cloud 워크스페이스
- 사이트 관리자 권한이 있는 Atlassian 계정
- AWS Security Agent Management Console에서 통합을 구성할 수 있는 권한

> **Important**
>
> Atlassian 사이트는 리전당 하나의 AWS 계정에만 연결할 수 있습니다. 동일한 Bitbucket 사이트를 동일한 리전 내의 다른 AWS 계정의 AWS Security Agent에 연결해야 하는 경우 먼저 기존 통합을 제거해야 합니다.

> **Note**
>
> Atlassian Forge 앱 요금은이 통합에 적용됩니다. 자세한 내용은 Atlassian 설명서의 Forge 플랫폼 요금을 참조하세요.

Bitbucket 연결 등록

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 추가(Add Integrations)를 선택합니다.
3. Bitbucket을 선택한 후 다음을 선택합니다.
4. 화면 지침에 따라 Bitbucket 워크스페이스에 AWS Security Agent Forge 앱을 설치합니다. 설치 후 설치 ID를 복사합니다. the section called “Atlassian 설치 ID 찾기”을(를) 참조하세요.
5. 설치 ID 필드에 Bitbucket에서 복사한 설치 ID를 붙여 넣습니다.
6. Bitbucket 워크스페이스 필드에 Bitbucket 워크스페이스 슬러그를 입력합니다. 예: `acme-corp`.
7. Authorize를 선택합니다.

   AWS Security Agent가 Bitbucket 워크스페이스에 액세스할 수 있도록 권한을 부여하기 위해 Atlassian으로 리디렉션됩니다. 권한 부여가 완료되면 콘솔로 돌아갑니다.
8. 세부 정보 등록 섹션에서이 연결의 등록 이름을 입력합니다. 유효한 문자는 문자, 숫자, 마침표, 밑줄 및 하이픈입니다.
9. 연결을 선택합니다.

> **Note**
>
> 연결 후 프로비저닝 지연
>
> 연결을 선택한 후 Atlassian 측에서 프로비저닝하는 데 몇 분 정도 걸릴 수 있습니다. 이 시간 동안 통합은 보류 중 상태를 보고합니다. 프로비저닝이 완료되기 전에 리포지토리를 선택하

거나 코드 검토를 활성화하려고 하면 아직 설치를 사용할 수 없다는 메시지가 표시될 수 있습니다. 몇 분 기다린 후 다시 시도하십시오.

Bitbucket 통합 문제 해결

Bitbucket 통합 프로세스 중에 문제가 발생하면 다음 지침을 사용하여 일반적인 문제를 해결합니다.

등록을 완료할 수 없음

등록 프로세스가 중단되면(브라우저 닫힘, 세션 제한 시간) Forge 앱이 Atlassian 사이트에 설치되지만 AWS 콘솔에는 등록되지 않을 수 있습니다.

해결 방법

- AWS Security Agent 콘솔로 돌아가서 통합 프로세스를 다시 시작합니다.
- Forge 앱은 설치된 상태로 유지되므로 다시 설치할 필요가 없습니다.

다른 AWS 계정에 이미 연결된 사이트

증상

- Atlassian 사이트가 이미 다른 계정과 연결되어 있음을 나타내는 오류

해결 방법

- Atlassian 사이트는 리전당 하나의 AWS 계정에만 연결할 수 있습니다.
- 기존 통합이 있는 AWS 계정을 식별하고 해당 계정을 사용하거나 기존 통합을 먼저 제거합니다.

설치를 사용할 수 없음

증상

- 리포지토리를 선택하거나 코드 검토를 활성화하면 Bitbucket 설치가 상태가 `PENDING_INSTALLATION`아니라 상태라는 메시지가 표시됩니다`AVAILABLE`.

해결 방법

- 설치는 여전히 Atlassian 측에서 프로비저닝 중입니다. 프로비저닝은 일반적으로 몇 분 내에 완료됩니다. 몇 분 정도 기다렸다가 다시 시도하세요.
- 몇 분 후에도 상태를 사용할 수 없는 경우 통합을 제거하고 다시 등록합니다. 다시 등록하기 전에 Bitbucket 워크스페이스에서 Forge 앱을 제거합니다. 지침은 Bitbucket 통합 제거를 참조하세요. 이전 Forge 앱을 제거하지 않고 다시 등록하면 설치가 보류 중 상태로 중단될 수 있습니다.

다음 단계

Bitbucket을 AWS Security Agent에 연결한 후:

- 이러한 리포지토리를 사용할 에이전트 공간으로 이동합니다.
- 코드 검토 활성화 또는 침투 테스트 설정을 선택하여 특정 리포지토리를 에이전트 스페이스에 연결
- AWS Security Agent가 취약성 수정과 함께 풀 요청을 제출할 수 있도록 코드 수정 활성화

Bitbucket 통합 제거

AWS Security Agent가 특정 Bitbucket 워크스페이스의 리포지토리에 더 이상 액세스할 필요가 없는 경우 Bitbucket 통합을 제거합니다.

제거를 위한 사전 조건

Bitbucket 통합을 제거하기 전에:

- 이 통합에서 연결된 리포지토리가 있는 에이전트 스페이스 확인
- 영향 이해:를 제거하면 연결된 모든 리포지토리에 대한 코드 검토, 침투 테스트 컨텍스트, 위협 모델링 및 자동 문제 해결이 중단됩니다.

1단계: AWS Security Agent에서 통합 제거

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 제거하려는 Bitbucket 통합을 찾습니다.
3. 통합을 선택합니다.
4. 제거를 선택합니다.

5. 확인 대화 상자를 검토하고 제거 확인을 선택합니다.

2단계: Forge 앱 제거

AWS Security Agent에서 통합을 제거한 후 Atlassian 사이트에서 Forge 앱을 제거합니다.

1. Bitbucket에서 Workspace 설정으로 이동합니다.
2. 앱 위조를 선택합니다.
3. AWS Security Agent로 연결을 찾습니다.
4. 제거를 선택합니다.

> **Important**
>
> Forge 앱이 자동으로 제거되지 않음
>
> AWS Security Agent 콘솔에서 통합을 제거해도 Atlassian 사이트에서 Forge 앱은 제거되지 않습니다. 동일한 Bitbucket 워크스페이스를 다시 등록하려는 경우 먼저 Forge 앱을 제거해야 합니다. 그렇지 않으면 새 설치가 보류 중 상태로 중단될 수 있습니다.

AWS Security Agent를 Confluence에 연결

AWS Security Agent를 Confluence Cloud에 연결하여 보안 평가를 위한 설명서 컨텍스트를 제공합니다. 코드 공급자와 달리 Confluence는 위협 모델, 아키텍처 문서, API 사양 및 보안 검토의 품질을 향상시키는 기타 자료를 제공하는 문서 소스 역할을 합니다. 시작하기 전에 the section called “Agent Spaces와의 통합 작동 방식”를 검토하여 에이전트 스페이스에서 등록을 재사용하는 방법을 이해합니다.

Confluence 통합은 여러 가지 용도로 사용됩니다.

- 설계 검토 컨텍스트 - 보안 설계 검토를 위한 아키텍처 문서 및 설계 사양 제공
- 위협 모델링 - 위협 분석을 위한 기존 위협 모델 및 시스템 설명서 제공
- 침투 테스트 컨텍스트 - 침투 테스트 중에 심층적으로 이해할 수 있도록 애플리케이션 설명서 제공

Confluence를 AWS Security Agent에 연결하려면 Atlassian 사이트에 AWS Security Agent Forge 앱을 설치하고 OAuth 권한 부여 흐름을 완료해야 합니다.

> **Note**
>
> AWS Security Agent는 Confluence 클라우드만 지원합니다. Confluence 데이터 센터 및 Confluence 서버는 지원되지 않습니다.

Confluence 통합 작동 방식

Confluence는 소스 코드 공급자가 아닌 설명서 공급자입니다. Forge 앱을 설치하고 AWS Management Console에 스페이스 또는 페이지를 연결한 후 AWS Security Agent는 Confluence 콘텐츠에 액세스하여 보안 평가 중에 컨텍스트를 제공할 수 있습니다.

AWS Security Agent는 페이지 콘텐츠를 읽어 애플리케이션 아키텍처, 보안 요구 사항 및 설계 결정을 이해합니다. 이 컨텍스트는 설계 검토, 코드 검토 및 침투 테스트 중에 보안 조사 결과의 품질과 관련성을 개선합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 관리자 액세스 권한이 있는 Confluence 클라우드 사이트
- 사이트 관리자 권한이 있는 Atlassian 계정
- AWS Security Agent Management Console에서 통합을 구성할 수 있는 권한

> **Important**
>
> Atlassian 사이트는 리전당 하나의 AWS 계정에만 연결할 수 있습니다. 동일한 Confluence 사이트를 동일한 리전 내의 다른 AWS 계정의 AWS Security Agent에 연결해야 하는 경우 먼저 기존 통합을 제거해야 합니다.

> **Note**
>
> Atlassian Forge 앱 요금은이 통합에 적용됩니다. 자세한 내용은 Atlassian 설명서의 Forge 플랫폼 요금을 참조하세요.

Confluence 연결 등록

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 통합 추가(Add Integrations)를 선택합니다.
3. Confluence를 선택한 후 다음을 선택합니다.
4. 화면의 지침에 따라 Atlassian 사이트에 AWS Security Agent Forge 앱을 설치합니다. 설치 후 설치 ID를 복사합니다. the section called “Atlassian 설치 ID 찾기”을(를) 참조하세요.
5. Confluence 사이트 URL 필드에와 같이 사이트 URL을 입력합니다`https://acme.atlassian.net`.
6. 설치 ID 필드에 Confluence에서 복사한 설치 ID를 붙여 넣습니다.
7. Authorize를 선택합니다.

   AWS Security Agent가 Confluence 사이트에 액세스할 수 있도록 권한을 부여하기 위해 Atlassian으로 리디렉션됩니다. 권한 부여가 완료되면 콘솔로 돌아갑니다.
8. 세부 정보 등록 섹션에서이 연결의 등록 이름을 입력합니다. 유효한 문자는 문자, 숫자, 마침표, 밑줄 및 하이픈입니다.
9. 연결을 선택합니다.

에이전트 스페이스에 대한 페이지 선택

Confluence 통합을 등록한 후 특정 페이지를 에이전트 스페이스에 연결합니다. 페이지를 선택하면 AWS Security Agent가 해당 페이지의 콘텐츠에 대한 읽기(가져오기) 액세스 권한을 부여합니다. 페이지당 기능 옵션은 없습니다. 에이전트는 연결된 모든 페이지를 읽습니다. 연결 마법사의 검토 단계에서 에이전트가 액세스하지 못하게 하려는 모든 페이지를 제거할 수 있습니다.

Confluence 통합 문제 해결

Confluence 통합 프로세스 중에 문제가 발생하면 다음 지침을 사용하여 일반적인 문제를 해결하세요.

등록을 완료할 수 없음

등록 프로세스가 중단되면 Forge 앱이 Atlassian 사이트에 설치되지만 AWS 콘솔에는 등록되지 않을 수 있습니다.

해결 방법

- AWS Security Agent 콘솔로 돌아가서 통합 프로세스를 다시 시작합니다.

- Forge 앱은 설치된 상태로 유지되므로 다시 설치할 필요가 없습니다.

Atlassian에서 제거된 Forge 앱

통합이 AWS Security Agent에 여전히 존재하는 동안 Atlassian 사이트에서 AWS Security Agent Forge 앱을 제거하는 경우:

증상

- 통합은 AWS 콘솔에 표시되지만 Confluence 콘텐츠에 액세스할 수 없습니다.
- 페이지를 나열하거나 읽으려고 할 때 발생하는 오류

해결 방법

- Atlassian Marketplace에서 Forge 앱 재설치
- 문제가 지속되면 AWS 콘솔에서 통합을 제거하고 다시 등록합니다.

다른 AWS 계정에 이미 연결된 사이트

해결 방법

- Atlassian 사이트는 리전당 하나의 AWS 계정에만 연결할 수 있습니다.
- 기존 통합이 있는 AWS 계정을 식별하고 해당 계정을 사용하거나 기존 통합을 먼저 제거합니다.

다음 단계

Confluence를 AWS Security Agent에 연결한 후:

- 이 설명서를 사용하려는 에이전트 공간으로 이동합니다.
- 설계 검토 및 침투 테스트를 위한 컨텍스트로 포함할 특정 페이지를 선택합니다.
- 필요한 경우 S3를 통해 추가 설명서 업로드(참조the section called “S3 버킷에서 에이전트 리소스 제공”)

Confluence 통합 제거

AWS Security Agent가 특정 Confluence 사이트의 설명서에 더 이상 액세스할 필요가 없는 경우 Confluence 통합을 제거합니다.

제거를 위한 사전 조건

Confluence 통합을 제거하기 전에:

- 이 통합에서 연결된 페이지가 있는 에이전트 스페이스 확인
- 영향 이해:를 제거하면이 통합의 콘텐츠를 사용하여 모든 에이전트 스페이스에서 설계 검토, 침투 테스트 및 위협 모델링에 대한 설명서 컨텍스트가 손상됩니다.

1단계: AWS Security Agent에서 통합 제거

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 제거하려는 Confluence 통합을 찾습니다.
3. 통합을 선택합니다.
4. 제거를 선택합니다.
5. 확인 대화 상자를 검토하고 제거 확인을 선택합니다.

2단계: Forge 앱 제거(선택 사항)

AWS Security Agent에서 통합을 제거한 후 Atlassian 사이트에서 Forge 앱을 선택적으로 제거할 수 있습니다.

1. Confluence에서 Confluence 관리로 이동합니다.
2. 앱을 선택합니다.
3. AWS Security Agent로 연결을 찾습니다.
4. 제거를 선택합니다.

프라이빗 호스팅 소스 제어에 연결

> **Important**
>
> 프라이빗 연결은 AWS Security Agent의 평가판 릴리스이며 변경될 수 있습니다.

AWS Security Agent는 GitLab 자체 관리형 인스턴스 및 GitHub Enterprise Server와 같은 프라이빗 네트워크에서 실행되는 소스 제어 시스템에 연결할 수 있습니다. 프라이빗 연결은 Amazon VPC Lattice

를 사용하여 시스템을 퍼블릭 인터넷에 노출하지 않고 AWS Security Agent와 프라이빗 인프라 간에 보안 연결을 설정합니다.

프라이빗 연결 작동 방식

프라이빗 연결은 AWS Security Agent와 VPC의 대상 리소스 간에 보안 네트워크 경로를 생성합니다. AWS Security Agent는 후드 아래에서 Amazon VPC Lattice를 사용하여이 연결을 설정합니다. VPC Lattice는 기본 네트워크 인프라를 관리하지 않고도 VPCs 및 모니터링하기 위한 AWS 애플리케이션 네트워킹 서비스입니다.

프라이빗 연결을 생성하는 경우:

1. 대상 서비스에 네트워크로 연결된 VPC, 서브넷 및 (선택 사항) 보안 그룹을 제공합니다.
2. AWS Security Agent는 서비스 관리형 리소스 게이트웨이를 생성하고 지정한 서브넷에 탄력적 네트워크 인터페이스(ENIs)를 프로비저닝합니다.
3. 에이전트는 리소스 게이트웨이를 사용하여 프라이빗 네트워크 경로를 통해 대상 서비스의 IP 주소 또는 DNS 이름으로 트래픽을 라우팅합니다.

서비스 관리형 및 자체 관리형 프라이빗 연결

AWS Security Agent는 프라이빗 연결에 대해 두 가지 모드를 지원합니다.

서비스 관리형(단순 설정의 경우 권장):

- AWS Security Agent는 VPC Lattice 리소스 게이트웨이를 자동으로 생성하고 관리합니다.
- 대상 서비스는 에이전트 스페이스가 생성된 동일한 AWS 계정에서 실행되어야 합니다.
- 여러 AWS 계정에 걸쳐 있을 수 없습니다.

자체 관리형(고급 또는 교차 계정 설정의 경우):

- 자체 VPC Lattice 리소스 구성을 생성하고 관리합니다.
- 기존 리소스 구성의 Amazon 리소스 이름(ARN)을 제공합니다.
- 교차 계정 액세스를 지원합니다(고객이 교차 계정 네트워킹을 관리함).

사전 조건

프라이빗 연결을 생성하기 전에 다음이 있는지 확인합니다.

- 활성 에이전트 공간
- 알려진 프라이빗 IP 주소 또는 DNS 이름에서 비공개로 연결할 수 있는 대상 서비스(GitLab 자체 관리형 또는 GitHub Enterprise Server)
- 대상 서비스는 최소 TLS 버전이 1.2인 HTTPS 트래픽을 제공해야 합니다.
- VPC의 서브넷(1~20개 서브넷). 고가용성을 위해 여러 가용 영역에서 서브넷을 선택하는 것이 좋습니다.
- (선택 사항) ENIs에 대한 트래픽을 제어하는 보안 그룹(최대 5개)

> **Note**
>
> VPC Lattice에서는 다음 가용 영역을 지원하지 않습니다`ilc1-az2`. `use1-az3`, , `usw1-az2`, `apne1-az3`, `apne2-az2euc1-az2`, `euw1-az4`, `cac1-az3`.

> **Note**
>
> 프라이빗 연결은 계정 수준 리소스입니다. 프라이빗 연결을 생성한 후 동일한 호스트에 연결해야 하는 여러 통합 및 에이전트 스페이스에서 프라이빗 연결을 재사용할 수 있습니다.

> **Note**
>
> OAuth 인증을 사용하는 공급자의 프라이빗 연결을 선택하면 프라이빗 연결이 공급자 엔드포인트와 토큰 교환 엔드포인트 모두에 적용됩니다. 프라이빗 연결이 두 엔드포인트로 트래픽을 라우팅할 수 있는 호스트 주소로 구성되어 있는지 확인합니다.

프라이빗 연결 생성

1. AWS Security Agent Management Console에서 통합으로 이동합니다.
2. 프라이빗 연결 탭을 선택합니다.
3. 프라이빗 연결 생성을 선택합니다.
4. 연결 세부 정보에서 이름을 입력합니다. 이름은 소문자, 숫자 및 하이픈을 포함할 수 있으며 문자 또는 숫자로 시작하고 끝나야 합니다.
5. 연결 세부 정보 섹션에서 AWS Security Agent가 대상 서비스에 연결하는 방법을 선택합니다.

- AWS Security Agent가 연결을 생성하고 관리하도록 하려면 기존 리소스 구성 사용을 선택 취소한 다음 리소스 위치, 액세스 제어 및 서비스 대상 세부 정보 섹션을 완료합니다.
- 이미 생성한 VPC Lattice 리소스 구성을 통해 라우팅하려면 기존 리소스 구성 사용을 선택한 다음 기존 리소스 구성 섹션을 완료합니다. 서비스 관리형 섹션은 적용되지 않습니다.

6. (서비스 관리형) 리소스 위치:
   a. 리소스가 있는 VPC를 선택합니다.
   b. 서브넷을 하나 이상 선택합니다. 두 개 이상의 가용 영역에서 서브넷을 선택하는 것이 좋습니다.
   c. (선택 사항) IP 주소 유형(IPv4, IPv6 또는 듀얼 스택)을 선택합니다.
7. (서비스 관리형) 액세스 제어:
   a. 연결된 리소스와 연결된 보안 그룹을 선택합니다.
   b. (선택 사항) 고급 구성에서 포트 범위 - 예를 들어 최대 11개의 쉼표로 구분된 TCP 포트 또는 범위를 입력합니다`443, 8080-8090`.
8. (서비스 관리형) 서비스 대상 세부 정보에서:
   a. 호스트 주소 필드에 대상 서비스의 DNS 호스트 이름 또는 IP 주소를 입력합니다.
   b. DNS 확인의 경우 공개적으로 확인 가능한 호스트 주소의 경우 퍼블릭을 선택하고 VPC(프라이빗 DNS)의 경우 VPC 내에서만 확인 가능한 주소를 선택합니다.
   c. (선택 사항) 대상이 자체 서명된 인증서 또는 프라이빗 인증 기관을 사용하는 경우 인증서 퍼블릭 키 필드에 PEM 인코딩 퍼블릭 키를 입력합니다. 이렇게 하면 AWS Security Agent가 TLS 연결을 신뢰할 수 있습니다.
9. (자체 관리형) 기존 리소스 구성의 리소스 구성에서 목록에서 VPC Lattice 리소스 구성을 선택하거나 리소스 구성 ID(로 시작`rcfg-`) 또는 해당 ARN을 입력합니다. 목록에는 현재 리전의 리소스 구성이 표시됩니다.
10. 연결 생성을 선택합니다.

    연결 상태가 생성 진행 중으로 변경됩니다. 이 프로세스는 최대 10분이 걸릴 수 있습니다. 완료되면 상태가 사용 가능으로 변경됩니다.

통합과 함께 프라이빗 연결 사용

GitLab 자체 관리형 또는 GitHub Enterprise Server 통합을 등록할 때 프라이빗 연결을 사용하여 엔드포인트에 연결을 선택한 다음 프라이빗 연결 목록에서 연결을 선택합니다. AWS Security Agent는 해당 연결을 통해 트래픽을 공급자에게 라우팅합니다. 사용 가능 상태의 연결만 목록에 표시됩니다.

등록 중에 프라이빗 연결 생성을 선택할 수도 있습니다. AWS Security Agent는 연결을 생성하는 동안 등록 초안을 보존한 다음 등록 흐름으로 돌아갑니다.

보안

- 퍼블릭 인터넷 노출 없음 - 모든 트래픽은 AWS 네트워크에 남아 있습니다.
- 고객 제어 보안 그룹 - 인바운드 및 아웃바운드 트래픽 규칙을 관리합니다.
- 최소 권한이 있는 서비스 연결 역할 - AWS Security Agent는 로 태그가 지정된 리소스로 범위가 지정된 서비스 연결 역할을 사용합니다`AWSSecurityAgentManaged`.

> **Note**
>
> 조직에 VPC Lattice API 작업을 제한하는 서비스 제어 정책(SCPs)이 있는 경우 서비스 연결 역할을 통해 서비스 관리형 리소스 게이트웨이가 생성됩니다. SCPs 서비스 연결 역할에 필요한 작업을 허용하는지 확인합니다.

프라이빗 연결 확인

프라이빗 연결이 사용 가능 상태에 도달하면 연결을 확인합니다.

- 대상 서비스가 실행 중이고 예상 포트에서 연결을 수락하는지 확인합니다.
- ENIs에 연결된 보안 그룹이 대상 포트에서 아웃바운드 트래픽을 허용하는지 확인합니다.
- 서비스의 보안 그룹이 VPC CIDR 범위 내의 VPC Lattice 데이터 영역 IPs로부터의 인바운드 트래픽을 허용하는지 확인
- 서브넷 라우팅 테이블이 ENI 서브넷과 서비스 간의 트래픽을 허용하는지 확인

프라이빗 연결 삭제

1. AWS Security Agent 콘솔에서 통합으로 이동합니다.
2. 프라이빗 연결 탭을 선택합니다.
3. 삭제할 프라이빗 연결을 선택합니다.
4. 삭제를 선택합니다.

> **Important**
>
> 통합에서 현재 사용 중인 프라이빗 연결은 삭제할 수 없습니다. 먼저 통합을 제거한 다음 프라이빗 연결을 삭제합니다.

다음 단계

- the section called “AWS Security Agent를 GitLab 자체 관리형에 연결” - 프라이빗 GitLab 인스턴스 연결
- the section called “AWS Security Agent를 GitHub Enterprise Server에 연결” - 프라이빗 GitHub Enterprise Server 연결

에이전트를 프라이빗 VPC 리소스에 연결

퍼블릭 인터넷에서 침투 테스트를 실행하려는 애플리케이션을 사용할 수 없는 경우 AWS Security Agent에 VPC 구성을 제공해야 합니다. AWS Security Agent는 VPC, 서브넷 및 보안 그룹을 포함한 이 VPC 구성을 사용하여 애플리케이션에 액세스합니다.

> **Note**
>
> 프라이빗 VPC에서 엔드포인트를 테스트할 때는 알려진 프라이빗 IPs 범위의 IP로 확인되는 엔드포인트만 허용됩니다(자세한 내용은 VPC CIDR 블록 참조). 다음 IPv4 및 IPv6 범위가 허용됩니다.
>
> ```
> 10.0.0.0/8
> 172.16.0.0/12
> 192.168.0.0/16
> fd00::/8
> ```

> **Note**
>
> 서브넷에 연결할 때 AWS Security Agent는 침투 테스트를 위해 구성된 서브넷에 ENI(Elastic Network Interface)를 생성합니다. 이 ENI에는 연결된 퍼블릭 IP 주소가 없으므로 퍼블릭 서브

넷의 VPC 인터넷 게이트웨이와 통신할 수 없습니다. 침투 테스트에 개방형 인터넷 액세스가 필요한 경우 연결된 VPC NAT 게이트웨이가 있는 프라이빗 서브넷을 대신 사용하십시오.

AWS Security Agent에 AWS 관리 콘솔에서 VPC에 대한 일반 액세스 권한을 부여합니다. Security Agent 웹 앱에서 사용자는 침투 테스트를 위한 특정 구성을 선택합니다.

에이전트 스페이스에 VPC를 추가하려면

1. 에이전트 스페이스 개요 페이지로 이동합니다.
2. 작업을 선택한 다음 침투 테스트 구성 편집을 선택합니다.
3. VPC 제목에서 VPC, 서브넷 및 보안 그룹을 지정합니다.

최대 5개의 VPCs.

필요한 에이전트 스페이스 서비스 역할 권한

VPC에서 침투 테스트를 실행하려면 에이전트 스페이스 서비스 역할에 다음 권한이 포함되어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeNetworkInterfaceAttribute",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": "ec2:CreateNetworkInterfacePermission",
      "Resource": "arn:aws:ec2:{{region}}:{{accountId}}:network-interface/*",
      "Condition": {
        "ArnEquals": {
          "ec2:Subnet": [
            "arn:aws:ec2:{{region}}:{{accountId}}:subnet/[[subnetIds]]"
          ]
        }
      }
    }
  ]
}
```

Security Agent 웹 앱에서 침투 테스트를 위한 특정 VPC 구성을 선택하려면

1. 침투 테스트 개요 페이지로 이동합니다.
2. VPC 구성을 추가해야 하는 침투 테스트를 선택한 다음 Pentest 세부 정보 수정을 선택합니다.
3. 다음을 선택하여 VPC 리소스 섹션으로 이동합니다.
4. VPC, 서브넷 및 보안 그룹 선택
5. 다음을 선택하여 마지막 섹션에 도달하고 침투 테스트를 저장합니다.

> **Note**
>
> 교차 계정 침투 테스트는 현재 AWS Resource Access Manager를 사용하여 공유되는 VPC 리소스(서브넷 및 보안 그룹)에 대해 지원됩니다. 인증 자격 증명에 사용되는 Secrets Manager 보안 암호 및 Lambda 함수는 AWS Security Agent 설정과 동일한 AWS 계정에서 구성해야 합니다.

다른 AWS 계정의 VPC 리소스에 대한 침투 테스트 실행

AWS Resource Access Manager를 사용하여 계정과 공유된 VPC 리소스에 대해 침투 테스트를 실행할 수 있습니다. 두 계정 모두 동일한 AWS Organization의 일부여야 합니다.

1. (선택 사항) AWS 조직에 대한 자동 리소스 공유 활성화

```
aws ram enable-sharing-with-aws-organization
```

2. VPC 리소스를 소유한 AWS 계정의 자격 증명을 사용하여 서브넷 및 보안 그룹 리소스를 침투 테스트 소유자 계정과 공유

```
aws ram create-resource-share \
    --name SharePentestResources \
    --resource-arns <subnet ARN> <security group ARN> \
    --principals <penetration test owner account ID>
```

3. 에이전트 스페이스 개요 페이지로 이동합니다.
4. 침투 테스트를 선택하고 서비스 역할 이름을 찾습니다.
5. IAM 역할이 공유 VPC 리소스에 대한 액세스 권한을 부여하는지 확인
6. 작업을 선택한 다음 침투 테스트 구성 편집을 선택합니다.
7. VPC 제목에서 공유 VPC, 서브넷 및 보안 그룹을 지정하고 업데이트된 구성을 저장합니다.
8. AWS Security Agent 웹 앱에서 침투 테스트 개요 페이지로 이동합니다.
9. VPC 구성을 추가해야 하는 침투 테스트를 선택한 다음 Pentest 세부 정보 수정을 선택합니다.
10. 공유 VPC 리소스를 사용하도록 침투 테스트 업데이트

기능 구성

문제 해결, S3 소스 및 대상 도메인을 포함하여 에이전트 스페이스에 대한 침투 테스트, 코드 검토 및 위협 모델링을 활성화하고 구성합니다.

침투 테스트 활성화

애플리케이션에서 자율 침투 테스트를 실행하도록 AWS Security Agent를 구성합니다. 이 설정을 통해 AWS Security Agent는 AWS 리소스에 액세스하고, 도메인 소유권을 확인하고, 웹 애플리케이션 및 APIs에서 악용 가능한 취약성을 식별하는 포괄적인 보안 테스트를 수행할 수 있습니다.

침투 테스트를 활성화하면 AWS Security Agent가 수동 테스트 지연 없이 애플리케이션 보안을 지속적으로 검증할 수 있습니다. 필요한 AWS 통합을 구성하면 AWS Security Agent가 퍼블릭 및 프라이빗 애플리케이션을 모두 테스트하고, 조사 결과를 CloudWatch에 기록하고, 인증된 테스트를 위해 자격 증명에 액세스할 수 있습니다.

이 절차에서는 대상 도메인을 구성하고, 선택적으로 테스트를 위해 VPC, CloudWatch 로깅, 자격 증명 스토리지 및 Lambda 함수를 설정하고, 추가 컨텍스트를 제공하도록 S3 통합을 구성하고, IAM 역할을 통해 서비스 액세스를 설정합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- IAM 역할을 생성할 수 있는 관리 권한이 있는 AWS 계정
- 도메인 소유권 확인 기능(DNS 또는 HTTP 레코드 수정)
- 테스트할 대상 도메인 또는 애플리케이션
- (선택 사항) 프라이빗 애플리케이션을 테스트하는 경우 VPC 구성 세부 정보
- (선택 사항) AWS Security Agent에 추가 아티팩트를 제공하는 경우 S3 버킷

1단계: 도메인 구성

마법사의 첫 번째 단계에서 테스트할 대상 도메인과 AWS Security Agent가 소유권을 확인하는 방법을 지정합니다.

1. 대상 도메인 섹션의 도메인 필드에 도메인을 입력합니다.
2. 확인 방법을 선택합니다.
   - DNS_TXT - 도메인의 DNS 구성에 TXT 레코드를 추가하여 도메인 소유권을 증명합니다.
   - HTTP_ROUTE - 도메인의 특정 URL에서 확인 파일을 호스팅하여 도메인 소유권을 증명합니다.
   - PRIVATE_VPC - 프라이빗 VPC 침투 테스트에만 사용할 수 있습니다. 도메인이 프라이빗 CIDR 범위의 IP로 확인되는지 확인합니다.
   - 자세한 내용은 the section called “침투 테스트를 위한 애플리케이션 도메인 활성화” 단원을 참조하십시오.
3. 다른 도메인 추가를 선택하여 도메인을 추가합니다(총 최대 5개).
4. 다음을 선택하여 도메인 확인을 진행합니다.

2단계: 도메인 확인

마법사의 두 번째 단계에서 구성한 각 도메인의 소유권을 확인합니다. AWS Security Agent는 침투 테스트를 수행하기 전에 확인된 소유권이 필요합니다.

1. 대상 도메인 테이블에 나열된 도메인을 검토합니다.
2. 각 도메인에 대해 해당 도메인을 선택하고 선택한 방법을 기반으로 확인을 트리거합니다.
   - Route 53 도메인(동일한 AWS 계정): 원클릭 확인을 선택합니다. AWS Security Agent는 DNS 레코드를 자동으로 생성하고 확인을 완료합니다.

- DNS TXT(다른 DNS 공급자): 확인 토큰을 복사하고 DNS 등록 기관에 TXT 레코드를 추가한 다음 도메인을 선택하고 확인을 선택합니다.
- HTTP 경로: 웹 서버의 필수 경로에 확인 토큰을 배치한 다음 도메인을 선택하고 확인을 선택합니다. 자세한 내용은 the section called “침투 테스트를 위한 애플리케이션 도메인 활성화”을 참조하세요.

3. 다음을 선택하여 선택적 구성으로 진행합니다.

> **Note**
>
> 확인된 도메인의 하위 도메인은 DNS TXT 또는 HTTP 경로 확인을 사용할 때 개별 확인이 필요하지 않습니다. 프라이빗 VPC 확인의 경우 대상 엔드포인트 도메인 이름은 확인을 위해 구성된 전체 도메인 이름과 일치해야 합니다.

> **Note**
>
> VPC 내의 프라이빗 도메인의 경우 도메인 확인 상태가 연결할 수 없음인 경우에도 진행할 수 있습니다. AWS Security Agent는 각 Pentest 실행이 시작될 때 프라이빗 엔드포인트에 대한 도메인 확인을 시도합니다.

3단계: (선택 사항) 추가 기능 구성

마법사의 세 번째 단계에서는 침투 테스트 범위를 확장하도록 선택적 AWS 리소스를 구성할 수 있습니다. 이 단계의 모든 섹션은 선택 사항이며 기본적으로 축소됩니다.

(선택 사항) VPC 설정 구성

VPC 내에서 호스팅되는 프라이빗 대상 도메인을 테스트하려는 경우 AWS Security Agent에 대한 VPC 설정을 구성합니다. 이 섹션은 선택 사항이며 기본적으로 축소됩니다.

1. VPCs.
2. VPC 드롭다운에서 프라이빗 대상 도메인을 호스팅하는 VPC를 선택합니다.
3. 서브넷 드롭다운에서 AWS Security Agent가 사용해야 하는 VPC 서브넷을 선택합니다.

> **Tip**
>
> 고가용성을 위해 여러 가용 영역에서 여러 서브넷을 선택합니다. 서브넷에 아웃바운드 연결을 위한 NAT 게이트웨이가 포함되어 있는지 확인합니다.

4. 보안 그룹 드롭다운에서 AWS Security Agent가 사용해야 하는 VPC 보안 그룹을 선택합니다.

> **Important**
>
> 보안 그룹이 AWS Security Agent가 침투 테스트를 수행할 수 있도록 아웃바운드 연결을 허용하는지 확인합니다.

(선택 사항) CloudWatch 로깅 구성

침투 테스트 실행의 로그를 저장하도록 CloudWatch를 구성합니다. 이 섹션은 선택 사항이며 기본적으로 축소됩니다.

1. CloudWatch 로그 섹션을 확장합니다.
2. 로그 그룹 드롭다운에서 기존 CloudWatch 로그 그룹을 선택합니다.
3. 로그 그룹을 선택하지 않으면 AWS Security Agent는 적절한 권한이 `/aws/securityagent/<agent name>/<pentest id>` 있는 라는 로그 그룹을 생성합니다.

> **Note**
>
> IAM 역할에 선택한 CloudWatch 로그 그룹에 쓸 수 있는 권한이 있는지 확인합니다.

(선택 사항) 테스트 자격 증명에 대한 보안 암호 구성

애플리케이션에 테스트용 인증 자격 증명이 필요한 경우 AWS Security Agent는 AWS Secrets Manager에서 해당 자격 증명을 안전하게 검색할 수 있습니다. 이 섹션은 선택 사항이며 기본적으로 축소됩니다.

1. 보안 암호 섹션을 확장합니다.
2. 애플리케이션의 자격 증명이 포함된 AWS Secrets Manager 보안 암호를 선택합니다.

3. 웹 애플리케이션에서 침투 테스트를 구성할 때 인증된 테스트를 위해 이러한 보안 암호를 참조할 수 있습니다.

> **Important**
>
> 자격 증명은 암호화되어 AWS Secrets Manager에 저장됩니다. IAM 역할에 AWS Security Agent용 Secrets Manager에 액세스하여 테스트 중에 이러한 자격 증명을 사용할 수 있는 권한이 있는지 확인합니다.

(선택 사항) 테스트 자격 증명에 대한 Lambda 함수 구성

테스트 중에 애플리케이션에 대한 자격 증명을 제공할 수 있는 Lambda 함수를 구성합니다. 이 섹션은 선택 사항이며 기본적으로 축소됩니다.

1. Lambda 함수 섹션을 확장합니다.
2. 애플리케이션에 인증 자격 증명을 제공할 수 있는 Lambda 함수를 선택합니다.
3. AWS Security Agent는 침투 테스트 중에 이러한 함수를 호출하여 동적으로 자격 증명을 얻습니다.

> **Note**
>
> IAM 역할에 지정된 Lambda 함수를 호출할 수 있는 권한이 있는지 확인합니다. Lambda 함수는 AWS Security Agent가 테스트 중에 사용할 수 있도록 예상 형식으로 자격 증명을 반환해야 합니다.

(선택 사항) S3 버킷 구성

AWS Security Agent에 입력으로 제공할 문서 또는 아티팩트를 업로드하려는 경우 S3 버킷 세부 정보를 제공합니다. 이 섹션은 선택 사항이며 기본적으로 축소됩니다.

1. S3 버킷 섹션을 확장합니다.
2. 버킷 필드에 S3 버킷 이름을 입력하거나 검색합니다.

> **Note**
>
> 나중에 GitHub 리포지토리를 연결하거나 웹 애플리케이션에서 직접 파일을 업로드할 수도 있습니다. 제공된 정보는 철저한 적용 범위를 보장하고, 오탐을 줄이고, 실행 가능한 결과를 제공할 수 있습니다.

(선택 사항) 서비스 액세스 구성

AWS Security Agent는 침투 테스트를 위해 AWS 리소스(VPC, CloudWatch 로그 그룹, Secrets Manager, Lambda 함수 등)에 액세스하려면 IAM 역할이 필요합니다. 이 섹션은 선택 사항이며 기본적으로 축소됩니다.

1. 서비스 액세스 섹션을 확장합니다.
2. 기본적으로 AWS Security Agent는 침투 테스트에 필요한 권한이 있는 기본 IAM 역할을 사용합니다.
3. IAM 역할을 사용자 지정하려면 다음 옵션 중 하나를 선택합니다.
   a. 기본 역할 생성 - AWS Security Agent는 필요한 권한을 가진 새 IAM 역할을 자동으로 생성합니다.
   b. 기존 서비스 역할 사용 - 드롭다운 메뉴에서 기존 IAM 역할을 선택합니다.
4. 기존 역할을 사용하는 경우:
   a. 기존 역할 선택에서 드롭다운 메뉴를 선택합니다.
   b. 목록에서 IAM 역할을 선택합니다.
   c. 필요한 경우 새로 고침 아이콘을 선택하여 목록을 업데이트합니다.

> **Note**
>
> 기본 IAM 역할에는 VPC 리소스, CloudWatch 로그 그룹, Secrets Manager, Lambda 함수 및 침투 테스트에 필요한 기타 서비스에 액세스할 수 있는 권한이 포함됩니다. 특정 보안 요구 사항이 없는 한 기본 IAM 역할을 사용하는 것이 좋습니다.

4단계: 침투 테스트 저장 및 활성화

필요한 모든 설정을 구성한 후 AWS Security Agent 에이전트에 대한 침투 테스트를 활성화합니다.

1. 모든 구성 섹션을 검토하여 정확성을 확인합니다.
2. 저장을 선택합니다.
3. AWS Security Agent는 구성을 검증하고 필요한 AWS 리소스를 생성합니다.

다음 단계

침투 테스트를 활성화한 후:

- AWS Security Agent 웹 애플리케이션에서 침투 테스트 범위 구성
- 테스트 결과에 대한 알림 기본 설정
- 침투 테스트 결과가 발견되면 검토 및 대응
- (선택 사항) 결과 수정을 위한 추가 리포지토리 구성

침투 테스트 실행 및 관리에 대한 자세한 내용은 AWS Security Agent 웹 애플리케이션 설명서를 참조하세요.

GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화

이제 풀 요청 코드 검토가 코드 검토 설정 마법사의 일부로 구성됩니다. GitHub 리포지토리를 에이전트 스페이스에 연결할 때 개별 리포지토리에 대한 풀 요청 설명을 활성화하여 AWS Security Agent가 풀 요청을 자동으로 분석하고 보안 조사 결과를 게시하도록 할 수 있습니다.

전체 설정 지침은 섹션을 참조하세요the section called “코드 검토 활성화”.

풀 요청 조사 결과가 GitHub에 표시되는 방식과 이에 응답하는 방법에 대한 자세한 내용은 섹션을 참조하세요the section called “풀 요청에서 코드 보안 조사 결과 검토”.

코드 검토 활성화

GitHub 리포지토리 또는 S3 버킷의 소스 코드를 연결하여 코드 검토를 활성화하도록 에이전트 스페이스를 구성합니다. 코드 검토는 소스 코드의 보안 취약성 및 조직의 사용자 지정 보안 요구 사항 준수를 분석합니다.

코드 검토 구성 설정은 에이전트 공간 전체의 작업입니다. 연결하는 통합 및 S3 버킷은 코드 검토 및 침투 테스트를 포함한 여러 기능에서 공유됩니다.

설정을 완료한 후 사용자는 AWS Security Agent 웹 애플리케이션에서 코드 검토를 생성하고 실행하여 리포지토리 및 S3 소스에서 보안 문제를 스캔할 수 있습니다.

> **Note**
>
> 이미 에이전트 스페이스에 연결된 GitHub 리포지토리가 있는 경우(예: 침투 테스트 설정을 통해) 코드 검토가 이미 활성화되어 있습니다. 이 설정을 건너뛰고 웹 애플리케이션으로 직접 이동하여 코드 검토를 생성할 수 있습니다. the section called “코드 검토 생성”을(를) 참조하세요.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Management Console에서 생성된 에이전트 공간(참조the section called “에이전트 스페이스 생성”)
- 다음 소스 코드 입력 중 하나 이상:
  - AWS Security Agent GitHub 앱이 설치된 GitHub 조직 또는 사용자 계정(참조the section called “AWS Security Agent를 GitHub 리포지토리에 연결”)
  - 검토하려는 소스 코드가 포함된 S3 버킷
- 에이전트 스페이스에 대한 통합을 구성할 수 있는 권한
- (선택 사항) 보안 요구 사항 검증을 사용하려는 경우 보안 요구 사항 팩에서 활성화된 하나 이상의 보안 요구 사항(참조the section called “보안 요구 사항 관리”)

코드 검토 설정 마법사 액세스

에이전트 스페이스의 코드 검토 설정으로 이동합니다.

1. AWS Security Agent 콘솔에서 에이전트 스페이스를 선택합니다.
2. 다음 위치 중 하나에서 코드 검토 활성화를 선택합니다.
   - 코드 검토 카드
   - 코드 검토 탭에서 코드 검토 활성화를 선택합니다.

> **Tip**
>
> AWS Security Agent가 코드 검토 피드백을 자동으로 처리하도록 하려면 코드 수정도 활성화합니다. 세부 정보는 the section called “사용자가 침투 테스트 및 코드 검토 결과의 수정을 시작할 수 있도록 지원” 섹션을 참조하세요.

설정 코드 검토 구성 마법사로 이동합니다.

1단계: 통합, 리포지토리 및 버킷 연결

마법사의 첫 번째 단계에서 소스 코드 입력을 연결하고 코드 검토 설정을 구성합니다. 계속하려면 하나 이상의 GitHub 리포지토리 또는 S3 버킷을 추가해야 합니다.

> **Important**
>
> 여기에 구성된 통합 및 S3 버킷은 에이전트 스페이스에서 공유됩니다. 변경 사항은 코드 검토 및 침투 테스트 기능 모두에 적용됩니다.

GitHub 리포지토리 연결

승인된 GitHub 조직 또는 사용자 계정에서 GitHub 리포지토리를 추가합니다. 추가를 선택하면 먼저 리포지토리를 선택한 다음 AWS Security Agent가 각 리포지토리에 대해 수행할 수 있는 작업을 구성하는 2단계 Connect GitHub 마법사가 열립니다.

1. 연결된 통합 섹션에서 추가를 선택합니다.
2. 검토하려는 리포지토리가 포함된 GitHub 등록을 선택합니다.
3. GitHub 리포지토리 연결 단계에서 연결하려는 각 리포지토리의 확인란을 선택합니다.
4. 다음을 선택하여 기능 관리로 이동합니다.

> **Note**
>
> 연결된 리포지토리는 코드 검토 및 침투 테스트 중에 코드 분석을 위해 읽기 전용으로 액세스됩니다. 다음 단계에서 검토 설명 게시 및 문제 해결 풀 요청 열기와 같은 쓰기 작업을 구성합니다.

> **Note**
>
> 아직 GitHub 통합을 등록하지 않은 경우 설정을 선택하여 AWS 보안 에이전트 GitHub 앱을 승인할 수 있는 통합 페이지로 이동합니다. 자세한 내용은 the section called “AWS Security Agent를 GitHub 리포지토리에 연결” 단원을 참조하십시오.

GitHub 리포지토리 기능 구성

GitHub 연결 마법사의 기능 관리 단계에서 각 리포지토리에서 AWS Security Agent가 수행할 수 있는 작업을 선택합니다. 코드 검토 설명 및 코드 수정은 리포지토리별로 독립적으로 설정됩니다.

1. 각 리포지토리에 대해의 코드 검토 설명을 전환하여 AWS Security Agent가 보안 조사 결과를 풀 요청에 대한 설명으로 게시하도록 합니다.
2. AWS Security Agent가 조사 결과를 수정할 수 있도록 각 리포지토리에 대해 코드 수정을 켭니다. 활성화되면 웹 앱 사용자는 조사 결과를 수정하는 풀 요청을 요청할 수 있으며, 풀 요청 작성자는 에이전트가 검토 의견을 처리`@AWS-Security-Agent fix all findings.`하도록 주석을 달 수 있습니다.
3. 저장을 선택하여 선택 사항을 적용하고 코드 검토 설정 마법사로 돌아갑니다.

리포지토리에 코드 검토 설명이 활성화된 경우:

- AWS Security Agent는 "검토 준비 완료"로 표시된 풀 요청을 자동으로 분석합니다. 초안 풀 요청은 분석되지 않습니다.
- 보안 조사 결과는 특정 수정 지침과 함께 풀 요청에 직접 검토 주석으로 게시됩니다.
- 분석은 구성된 코드 검토 설정(보안 취약성, 사용자 지정 요구 사항 또는 둘 다)을 사용합니다.

> **Note**
>
> 풀 요청 주석은 프라이빗 GitHub 리포지토리에만 사용할 수 있습니다.

리포지토리에 코드 수정이 활성화되면 웹 앱 사용자는 해당 리포지토리의 코드 검토 및 침투 테스트 결과에 대한 수정을 시작할 수 있으며 AWS Security Agent는 각 수정을 풀 요청으로 제공합니다. 자세한 내용은 the section called “사용자가 침투 테스트 및 코드 검토 결과의 수정을 시작할 수 있도록 지원” 단원을 참조하십시오.

풀 요청 조사 결과가 GitHub에 표시되는 방법과 이에 응답하는 방법에 대한 자세한 내용은 섹션을 참조하세요the section called “풀 요청에서 코드 보안 조사 결과 검토”.

S3 버킷 추가

코드 검토를 위해 소스 코드 또는 컨텍스트 리소스가 포함된 S3 버킷을 추가합니다.

1. S3 버킷 섹션에서 S3 리소스 추가를 선택합니다.

2. 소스 코드가 포함된 버킷 또는 접두사에 대한 S3 URI를 입력합니다.

3. 추가를 선택합니다.

> **Note**
>
> 최대 10개의 S3 리소스를 추가할 수 있습니다. S3 버킷은 코드 검토 및 침투 테스트를 포함한 여러 기능에서 공유됩니다.

> **Tip**
>
> AWS Security Agent가 보안 문제를 분석할 소스 코드, 구성 파일, infrastructure-as-code 템플릿 또는 기타 아티팩트가 포함된 S3 버킷을 추가할 수 있습니다.

코드 검토 설정 구성

AWS Security Agent가 코드 검토 중에 분석하는 보안 문제의 유형을 구성합니다. 이 설정은이 에이전트 스페이스에서 코드 검토가 활성화된 모든 리포지토리 및 소스에 적용됩니다.

1. 코드 검토 설정 섹션에서 다음 옵션 중 하나를 선택합니다.
   - 보안 요구 사항 검증 - 코드가 보안 요구 사항 팩에서 활성화한 보안 요구 사항을 준수하는지 확인합니다.
   - 보안 취약성 조사 결과 - 코드에서 일반적인 보안 취약성을 식별합니다.
   - 보안 요구 사항 및 취약성 조사 결과 - 활성화한 보안 요구 사항과 일반적인 보안 취약성을 모두 준수하는지 코드를 분석합니다. 기본 설정입니다.

> **Note**
>
> 보안 요구 사항 검증이 활성화되면 AWS Security Agent는 보안 요구 사항 팩에서 활성화한 보안 요구 사항을 기준으로 코드를 확인합니다. 보안 요구 사항 검증을 선택했지만 하나 이상의 보안 요구 사항이 활성화되지 않은 경우 AWS Security Agent는 요구 사항 기반 조사 결과를 식별하지 않습니다. 보안 요건에 대한 자세한 내용은 the section called “보안 요구 사항 관리”을 참조하십시오.

1. 다음을 선택하여 선택적 구성으로 진행합니다.

2단계: 선택적 구성

마법사의 두 번째 단계에서 코드 검토 환경에 대한 선택적 CloudWatch 로깅 및 서비스 액세스 설정을 구성합니다.

(선택 사항) CloudWatch 로그 구성

코드 검토 중에 애플리케이션 동작을 캡처하고 분석하도록 CloudWatch 로그 그룹을 구성합니다.

1. CloudWatch 로그 섹션을 확장합니다.
2. 로그 그룹 드롭다운의 AWS 계정에서 기존 CloudWatch 로그 그룹을 하나 이상 선택합니다.

> **Note**
>
> 로그 그룹을 선택하지 않으면 AWS Security Agent는 코드 검토 실행 로그를 저장할 기본 로그 그룹을 자동으로 생성합니다.

(선택 사항) 서비스 액세스 구성

AWS Security Agent가 코드 검토를 위해 S3 버킷 및 CloudWatch 로그와 같은 AWS 리소스에 액세스하는 데 사용하는 IAM 서비스 역할을 구성합니다.

1. 서비스 액세스 섹션을 확장합니다.
2. 다음 옵션 중 하나를 선택합니다.
   - 기본 역할 생성 - AWS Security Agent는 코드 검토에 필요한 권한을 가진 새 IAM 역할을 자동으로 생성합니다.
   - 기존 서비스 역할 사용 - 드롭다운 메뉴에서 기존 IAM 역할을 선택합니다.
3. 기본 역할을 사용하는 경우 서비스 역할 이름을 입력합니다. 이름은 계정의 모든 역할에서 고유해야 하며 영숫자와 문자를 사용해야 `+=,.@-_` 하며 공백을 포함할 수 없습니다.

> **Note**
>
> 서비스 역할 이름의 최대 길이는 64자입니다. 기존 역할을 선택하지 않으면 서비스 역할이 자동으로 생성됩니다.

1. 저장을 선택하여 설정을 완료합니다.

설정 후

코드 검토 설정 마법사를 완료한 후:

- 에이전트 스페이스 페이지의 코드 검토 카드에 준비됨 상태가 표시됩니다.
- 사용자는 웹 애플리케이션을 시작하고 코드 검토를 생성하여 연결된 리포지토리 및 S3 소스를 스캔할 수 있습니다.
- 코드 검토 탭에서 구성 편집을 선택하여 언제든지 구성을 수정할 수 있습니다.

코드 검토 구성 편집

초기 설정 후 코드 검토 구성을 수정하려면:

1. AWS Security Agent 콘솔에서 에이전트 공간으로 이동합니다.
2. 코드 검토 탭을 선택합니다.
3. 구성 편집을 선택합니다.
4. 필요에 따라 통합, S3 버킷, 코드 검토 설정, CloudWatch 로그 또는 서비스 액세스를 업데이트합니다.
5. 저장을 선택합니다.

다음 단계

코드 검토 구성을 설정한 후:

- 웹 애플리케이션을 시작하여 코드 검토를 생성하고 실행합니다(참조the section called “코드 검토 생성”).
- 코드베이스가 증가함에 따라 추가 GitHub 리포지토리 또는 S3 버킷 연결
- 조직별 검증을 위한 보안 요구 사항 팩 구성(참조the section called “보안 요구 사항 관리”)

- 풀 요청 조사 결과가 GitHub에 어떻게 표시되는지 검토(참조the section called “풀 요청에서 코드 보안 조사 결과 검토”)

위협 모델링 활성화

소스 코드 리포지토리를 연결하고 AWS 리소스를 구성하여 위협 모델링을 활성화하도록 에이전트 스페이스를 구성합니다. 위협 모델링은 애플리케이션의 아키텍처를 분석하고 소스 코드, 설계 문서 또는 둘 다에서 보안 위협을 식별합니다.

위협 모델링 구성 설정은 에이전트 공간 전반의 작업입니다. 연결하는 통합 및 S3 버킷은 위협 모델링, 코드 검토, 침투 테스트를 비롯한 여러 기능에서 공유됩니다.

설정을 완료한 후 사용자는 AWS Security Agent 웹 애플리케이션에서 위협 모델을 생성하고 실행할 수 있습니다.

> **Note**
>
> 에이전트 스페이스에 연결된 리포지토리 또는 S3 버킷이 이미 있는 경우(예: 코드 검토 또는 침투 테스트 설정을 통해) 위협 모델링이 이미 활성화되어 있을 수 있습니다. 웹 애플리케이션으로 직접 이동하여 위협 모델을 생성할 수 있습니다. the section called “위협 모델 생성”을(를) 참조하세요.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Management Console에서 생성된 에이전트 공간(참조the section called “에이전트 스페이스 생성”)
- 에이전트 스페이스에 대한 통합을 구성할 수 있는 권한
- (선택 사항) AWS Security Agent 앱이 설치된 GitHub, GitLab 또는 Bitbucket 조직(AWS Security Agent를 GitHub 리포지토리에 연결, AWS Security Agent를 GitLab 리포지토리에 연결 또는 AWS Security Agent를 Bitbucket 리포지토리에 연결 참조)

위협 모델링 설정 마법사 액세스

에이전트 스페이스의 위협 모델링 구성으로 이동합니다.

1. AWS Security Agent 콘솔에서 에이전트 스페이스를 선택합니다.
2. 위협 모델 카드 또는 위협 모델 탭에서 위협 모델 구성을 선택합니다.

위협 모델 구성 마법사로 이동합니다.이 마법사에는 두 가지 선택적 단계가 있습니다.

1단계: 소스 코드 리포지토리 연결(선택 사항)

위협 모델링을 활성화하려는 GitHub, GitLab 또는 Bitbucket 리포지토리를 연결합니다. 위협 모델 자체는 웹 애플리케이션에서 생성되고 표시됩니다.

> **Important**
>
> 여기에 구성된 통합은 에이전트 스페이스에서 공유됩니다. 변경 사항은 위협 모델링, 코드 검토 및 침투 테스트 기능에 적용됩니다.

리포지토리 연결

1. 연결된 통합 섹션에서 추가를 선택합니다.
2. 사용하려는 리포지토리가 포함된 등록을 선택합니다.
3. 연결하려는 각 리포지토리의 확인란을 선택합니다.
4. 저장을 선택하여 선택 사항을 적용합니다.

> **Note**
>
> 아직 통합을 등록하지 않은 경우 설정을 선택하여 AWS Security Agent 앱을 승인할 수 있는 통합 페이지로 이동합니다. 자세한 내용은 GitHub 리포지토리에 AWS Security Agent 연결, GitLab 리포지토리에 AWS Security Agent 연결 또는 Bitbucket 리포지토리에 AWS Security Agent 연결을 참조하세요.

1. 다음을 선택하여 선택적 구성으로 진행합니다.

2단계: 선택적 구성

위협 모델링 환경에 대한 S3 버킷, CloudWatch 로깅 및 서비스 액세스 설정을 구성합니다. 이러한 설정은 에이전트 스페이스의 다른 기능과 공유됩니다.

S3 버킷(선택 사항)

위협 모델링 중에 에이전트가 컨텍스트로 사용할 소스 코드가 포함된 S3 버킷을 추가합니다.

1. S3 버킷 섹션에서 S3 리소스 추가를 선택합니다.
2. 버킷 또는 접두사에 대한 S3 URI를 입력합니다.
3. 추가를 선택합니다.

> **Note**
>
> 최대 10개의 S3 리소스를 추가할 수 있습니다.

CloudWatch 로그(선택 사항)

위협 모델 실행 중에 애플리케이션 동작을 캡처하고 분석하도록 CloudWatch 로그 그룹을 구성합니다.

1. CloudWatch 로그 섹션에서 AWS 계정의 기존 CloudWatch 로그 그룹을 하나 이상 선택합니다.

서비스 액세스

AWS Security Agent가 위협 모델링을 위해 S3 버킷 및 CloudWatch 로그와 같은 AWS 리소스에 액세스하는 데 사용하는 IAM 서비스 역할을 구성합니다. 위협 모델링을 활성화하려면 서비스 역할이 필요합니다.

1. 서비스 액세스 섹션의 드롭다운에서 기존 IAM 서비스 역할을 선택하거나 서비스 역할이 자동으로 생성되도록 비워 둡니다.

> **Note**
>
> 기존 역할을 선택하지 않으면 서비스 역할이 자동으로 생성됩니다.

1. 저장을 선택하여 구성을 완료합니다.

설정 후

위협 모델링 구성을 완료한 후:

- 에이전트 스페이스 페이지의 위협 모델 카드에 준비됨 상태가 표시됩니다.
- 사용자는 웹 애플리케이션을 시작하고 연결된 소스 코드, 업로드된 범위 문서, Confluence 페이지 또는 이러한 입력의 조합에서 위협 모델을 생성할 수 있습니다.

다음 단계

위협 모델링을 활성화한 후:

- 웹 애플리케이션을 시작하여 위협 모델을 생성하고 실행합니다(위협 모델 생성 참조).
- 코드베이스가 증가함에 따라 추가 리포지토리 또는 S3 버킷 연결
- Confluence를 연결하여 페이지를 범위 문서로 선택 활성화(Confluence에 AWS 보안 에이전트 연결 참조)

사용자가 침투 테스트 및 코드 검토 결과의 수정을 시작할 수 있도록 지원

AWS Management Console에서 자동 문제 해결을 활성화하면 AWS Security Agent 웹 앱 사용자가 특정 결과에 대한 수정을 요청할 수 있습니다. AWS Security Agent는 영향을 받는 리포지토리에서 각 수정 사항을 GitHub 풀 요청으로 제공합니다.

에이전트 공간 내에서 리포지토리당 문제 해결을 활성화합니다. 침투 테스트 및 코드 검토 탭은 동일한 리포지토리 구성 마법사를 열어 두 탭 중 하나를 사용하여 자동 문제 해결 활성화 설정을 관리할 수 있습니다. 문제 해결은 두 기능의 결과에 모두 적용되므로 리포지토리당 한 번만 활성화하면 됩니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

1. 활성화된 침투 테스트(참조the section called “침투 테스트 활성화”) 또는 코드 검토(참조the section called “코드 검토 활성화”)
2. GitHub 조직에 대한 AWS 보안 에이전트 GitHub 앱을 설치하고 권한을 부여했습니다(참조the section called “AWS Security Agent를 GitHub 리포지토리에 연결”).

리포지토리 구성 마법사 열기

리포지토리 설정 방식과 일치하는 진입점을 선택합니다.

침투 테스트 탭에서

이 경로를 사용하여 침투 테스트를 위한 GitHub 리포지토리를 추가하고 동일한 패스에서 문제 해결을 구성합니다.

1. 에이전트 스페이스 개요 페이지로 이동합니다.
2. 침투 테스트 탭을 선택합니다.
3. 리포지토리를 소유한 GitHub 등록을 선택합니다.
   a. GitHub 등록을 에이전트 스페이스와 연결하지 않은 경우 AWS 보안 에이전트에 GitHub 연결 정보 상자에서 추가를 선택하여 등록을 선택합니다.
   b. 하나 이상의 GitHub 등록이 이미 연결되어 있는 경우 연결된 통합 섹션에서 추가를 선택하여 다른 등록을 선택합니다.
4. 다음을 선택하여 GitHub 리포지토리를 선택합니다.
5. 다음을 선택하여 리포지토리 기능을 구성합니다.

코드 검토 탭에서

리포지토리가 코드 검토를 위해 이미 연결되어 있는 경우이 경로를 사용합니다.

1. 에이전트 스페이스 개요 페이지로 이동합니다.
2. 코드 검토 탭을 선택합니다.
3. GitHub 리포지토리 테이블에서 편집을 선택하여 리포지토리 구성 마법사를 엽니다.

자동 문제 해결 활성화 및 저장

리포지토리 기능에 도달하면 위의 두 경로 중 하나를 단계별로 진행합니다.

1. 자동 수정 활성화됨 열에서 수정하려는 각 리포지토리를 활성화됨으로 표시합니다.
2. 연결을 선택하여 구성을 저장합니다.

이제 웹 앱 사용자는 이러한 리포지토리의 결과에 대한 문제 해결을 시작할 수 있으며 AWS Security Agent는 제안된 수정 사항을 사용하여 풀 요청을 엽니다.

S3 버킷에서 에이전트 리소스 제공

AWS Security Agent에 침투 테스트를 지원하는 API 문서, 위협 모델 문서 및 소스 코드와 같은 S3 버킷의 리소스에 대한 액세스 권한을 부여할 수 있습니다.

AWS Security Agent에 AWS Management Console의 S3 버킷에 대한 읽기 액세스 권한을 부여합니다. Security Agent 웹 앱에서 사용자는 필요에 따라 S3에서 개별 리소스를 선택합니다.

1. 에이전트 스페이스 개요 페이지로 이동합니다.
2. 작업을 선택한 다음 침투 테스트 구성 편집을 선택합니다.
3. S3 버킷 섹션에서 S3 버킷이 포함된 S3 URI를 정의합니다. S3 여러 S3 버킷을 추가할 수 있습니다.

침투 테스트를 위한 애플리케이션 도메인 활성화

애플리케이션에서 침투 테스트를 실행하려면 먼저 대상 도메인을 추가하고 소유권을 확인해야 합니다. AWS Security Agent는 확인된 도메인에 대해서만 침투 테스트를 수행합니다.

> **Note**
>
> 애플리케이션에서 사용할 수 있는 보조 도메인은 검증할 필요가 없습니다. 침투 테스트를 적극적으로 실행할 도메인을 검증하기만 하면 됩니다.

1단계: 대상 도메인 추가

대상 도메인을 추가하려면 에이전트 스페이스 개요 페이지에서 침투 테스트 탭으로 이동합니다. 침투 테스트를 이미 구성했는지 여부에 따라 다음 방법 중 하나를 사용합니다.

- 최초 설정: 침투 테스트 설정을 선택하여 침투 테스트 마법사를 엽니다. 첫 번째 단계에서 도메인을 입력하고 확인 방법을 선택합니다.
- 기존 구성에 도메인 추가: 대상 도메인 테이블에서 도메인 추가를 선택합니다. 도메인을 입력하고 모달에서 확인 방법을 선택합니다.

또는와 같은 기본 도메인 또는 하위 도메인을 추가할 수 있습니다`example.com`. AWS는 TXT 레코드를 생성할 권한이 있는 하위 도메인을 사용할 것을 제안합니다.

> **Note**
>
> DNS TXT 또는 HTTP 경로 확인을 사용하여 도메인을 확인하면 해당 도메인의 하위 도메인이 자동으로 포함됩니다. 예를 들어 확인을 `example.com` 통해 추가 확인

`billing.example.com` 없이 `api.example.com` 또는를 테스트할 수 있습니다. 프라이빗 VPC 확인의 경우 대상 엔드포인트 도메인 이름은 확인을 위해 구성된 전체 도메인 이름과 일치해야 합니다.

다음 확인 방법 중 하나를 선택합니다.

- DNS TXT 레코드: DNS 공급자를 통해 DNS TXT 레코드를 생성하여 도메인 소유권을 증명합니다.
- HTTP 경로: AWS Security Agent에서 제공하는 고유한 토큰이 포함된 경로를 웹 서버에 생성하여 도메인 소유권을 증명합니다.
- 프라이빗 VPC: 프라이빗 VPC 침투 테스트에만 사용할 수 있습니다. 도메인이 프라이빗 CIDR 범위의 IP로 확인되는지 확인합니다. 구성된 도메인 이름은 연결된 대상 엔드포인트의 전체 도메인 이름과 일치해야 합니다.

2단계: 도메인 소유권 확인

도메인을 추가한 후 선택한 방법을 사용하여 소유권을 확인합니다. 도메인을 선택하고 확인을 선택하여 대상 도메인 테이블에서 언제든지 확인을 트리거할 수 있습니다.

DNS TXT 레코드를 사용하여 확인

AWS Security Agent는 TXT DNS 레코드 값을 생성합니다. 소유권을 증명하려면 DNS 공급자에이 레코드를 추가해야 합니다.

도메인이 Route 53(동일한 AWS 계정)에 등록된 경우:

AWS Security Agent는 DNS 레코드를 자동으로 생성할 수 있습니다. 대상 도메인 테이블에서 도메인을 선택하고 원클릭 확인을 선택합니다. AWS Security Agent는 DNS 검증 레코드를 생성하고 확인을 자동으로 완료합니다.

도메인이 다른 DNS 공급자에 등록된 경우:

1. AWS Security Agent에서 제공하는 TXT 레코드 값을 복사합니다.
2. DNS 등록 대행자와 함께 TXT 레코드를 추가합니다.
3. 대상 도메인 테이블로 돌아가 도메인을 선택하고 확인을 선택합니다.

HTTP 라우팅 검증을 사용하여 확인

이 방법은 웹 서버에 고유한 토큰을 배치하여 도메인 소유권을 증명합니다. 도메인 소유자 또는 승인된 웹 관리자만 웹 서버에서 경로를 생성할 수 있으며, 이는 소유권을 증명합니다.

1. 웹 서버의 다음 경로에 파일을 생성합니다.

```
.well-known/aws/securityagent-domain-verification.json
```

2. AWS Security Agent에서 제공한 토큰을 다음 형식을 사용하여 파일에 넣습니다.

```
{
  "tokens": ["<insert-token>"]
}
```

3. 대상 도메인 테이블로 돌아가 도메인을 선택하고 확인을 선택합니다. AWS Security Agent는 HTTPS GET 요청을 확인 URL로 보내고 토큰을 검증합니다.
4. 퍼블릭 인터넷에서 도메인에 액세스할 수 있는 경우 확인을 실행하기 전에 도메인에 유효한 SSL 인증서가 있는지 확인합니다.

> **Note**
>
> 도메인이 여러 에이전트 공간에 등록되어 있고 HTTP 라우팅 검증을 사용하는 경우 두 에이전트 공간에 대한 토큰을 동일한 `tokens` 배열에 배치할 수 있습니다.

도메인 소유권 확인 우회

엔드포인트에서 침투 테스트를 수행할 권한이 있지만 소유권 확인을 완료할 수 없는 고객은 당사에 수동 확인을 요청할 수 있습니다. 이 요청을 하려면 고객 지원 사례를 여십시오. 요청에는 비즈니스 사용 사례와 수동 소유권 확인에 대한 근거(즉, 엔드포인트에서 침투 테스트를 수행할 권한이 있는 이유)가 포함되어야 합니다.

침투 테스트에 사용되는 관리형 대상 도메인

AWS Management Console에서 침투 테스트를 위해 에이전트 공간에서 사용하는 대상 도메인을 추가하고 관리할 수 있습니다. 침투 테스트에 사용하기 전에 이러한 대상 도메인을 확인해야 합니다. 대상 도메인 확인에 대한 자세한 내용은 섹션을 참조하세요. the section called “침투 테스트 활성화”

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

1. 활성화된 침투 테스트(참조the section called “침투 테스트 활성화”)

대상 도메인 리소스 관리

- 대상 도메인 개요 페이지로 이동합니다.
- 계정과 연결된 모든 대상 도메인 리소스가 표시되어야 합니다.
  - 계정과 연결된 대상 도메인이 보이지 않는 경우의 단계에 따라 대상 도메인을 the section called “침투 테스트 활성화” 생성하고 확인합니다.
- 대상 도메인은 에이전트 공간 간에 재사용하고 확인 상태를 공유할 수 있습니다.
  - 에이전트 공간에 기존 대상 도메인을 추가하려면 에이전트 공간의 침투 테스트 탭으로 이동합니다. 도메인 이름 필드의 이전에 등록된 사용 가능한 도메인 중에서 선택에서 도메인 추가를 선택하고 원하는 도메인을 선택합니다.
  - 대상 도메인을 침투 테스트에 사용하려면 먼저 에이전트 공간과 연결해야 합니다.
- 에이전트 공간에서 도메인을 제거해도 도메인은 삭제되지 않습니다. 대상 도메인 개요 페이지에서 연결된 대상 도메인을 영구적으로 삭제할 수 있습니다.

대상 도메인 확인

대상 도메인을 침투 테스트에 사용하려면 먼저 다음 방법 중 하나를 사용하여 확인해야 합니다.

- Route 53 도메인(동일한 AWS 계정): 원클릭 확인을 선택합니다. AWS Security Agent는 DNS 레코드를 자동으로 생성하고 확인을 완료합니다.
- DNS TXT(다른 DNS 공급자): 확인 토큰을 복사하고 DNS 등록 기관에 TXT 레코드를 추가한 다음 도메인을 선택하고 확인을 선택합니다.
- HTTP 경로: 웹 서버의 필수 경로에 확인 토큰을 배치한 다음 도메인을 선택하고 확인을 선택합니다. 자세한 내용은 the section called “침투 테스트를 위한 애플리케이션 도메인 활성화”을 참조하세요.
- 프라이빗 VPC: 대상 도메인의 IP가 프라이빗 CIDR 범위에 속하는지 확인합니다(프라the section called “에이전트를 프라이빗 VPC 리소스에 연결”이빗 CIDR 범위 목록은 참조). 침투 테스트 대상 엔드포인트 도메인 이름은 확인을 위해 구성된 전체 도메인 이름과 일치해야 합니다. 프라이빗 VPC 침투 테스트에만 사용 가능

> **Note**
>
> DNS TXT, HTTP 경로 및 Route 53 확인의 경우 확인된 도메인의 하위 도메인이 자동으로 포함되며 별도의 확인이 필요하지 않습니다. 프라이빗 VPC 확인의 경우 각 도메인을 개별적으로 확인해야 합니다.

보안 요구 사항 관리

AWS 관리형 팩, 사용자 지정 팩 또는 자체 설명서에서 가져온 요구 사항을 사용하여 AWS Security Agent가 설계 및 코드 검토 중에 평가하는 보안 요구 사항을 정의합니다.

보안 요구 사항 관리

AWS Security Agent가 애플리케이션을 보안 요구 사항 팩으로 분석하는 데 사용하는 보안 요구 사항을 구성합니다. 팩은 보안 요구 사항의 모음입니다. 보안 설명서를 업로드하여 AWS관리형 팩을 활성화하고, 사용자 지정 팩을 생성하고, 사용자 지정 팩에 대한 요구 사항을 생성할 수 있습니다.

개요

보안 요구 사항은 AWS Security Agent가 설계 검토 및 코드 검토 중에 애플리케이션을 평가하는 보안 표준 또는 정책을 정의합니다. 설계 또는 코드가 요구 사항을 충족하지 않는 경우 AWS Security Agent는 조사 결과를 보고합니다. 모든 보안 요구 사항은 보안 요구 사항 팩에 속합니다. 보안 요구 사항은 모든 에이전트 스페이스에서 공유되며 설계 및 코드 검토 중에 평가됩니다.

AWS Security Agent는 별도의 탭에 표시된 두 가지 유형의 팩을 제공합니다.

- 관리형 보안 요구 사항 팩 AWS- 업계 표준 및 모범 사례를 기반으로 제공되는 보안 요구 사항 팩입니다. 이러한 팩을 즉시 활성화하여 사용자 지정 구성 없이 일반적인 보안 정책을 기준으로 평가할 수 있습니다. 관리형 팩의 요구 사항은 읽기 전용입니다. AWS관리형 요구 사항을 사용자 지정 요구 사항의 템플릿으로 사용할 수 있습니다. 사용 가능한 관리형 팩 목록은 AWS 관리형 보안 요구 사항 팩을 참조하세요.
- 사용자 지정 보안 요구 사항 팩 - 조직의 특정 정책 및 표준을 적용하기 위해 생성하는 팩입니다. 처음부터 사용자 지정 팩에 요구 사항을 빌드하거나, 사용자 지정 AWS관리형 요구 사항을 시작점으로 구축하거나, 보안 설명서를 업로드하여 요구 사항을 생성합니다.

> **Note**
>
> 규정 준수 프레임워크 팩은 특정 프레임워크 규정 준수와 관련이 있을 수 있는 보안 요구 사항을 식별하는 데 도움이 되도록 설계되었습니다. 규정 준수를 평가하거나 감사를 통과할 것이라고 보장하지 않습니다.

팩을 단위로 활성화 및 비활성화합니다. 팩이 활성화되면 AWS Security Agent는 설계 및 코드 검토 중에 해당 팩의 요구 사항을 기준으로 애플리케이션을 평가합니다.

> **Note**
>
> 팩 활성화 또는 비활성화는 새로운 설계 검토 및 코드 검토에 적용됩니다. 기존 검토는 영향을 받지 않습니다.

관리형 팩 활성화 또는 비활성화

AWS관리형 팩을 활성화하여 일련의 AWS유지 관리형 보안 요구 사항을 기준으로 애플리케이션을 평가합니다. 더 이상 필요하지 않은 경우 비활성화합니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 관리형 보안 요구 사항 팩 탭을 선택합니다.
4. 활성화 또는 비활성화할 팩을 선택합니다.
5. 다음 중 하나를 수행하세요.
   a. 팩을 활성화하려면 팩 활성화를 선택합니다.
   b. 팩을 비활성화하려면 팩 비활성화를 선택합니다.

> **Tip**
>
> 팩을 선택하여 팩에 포함된 보안 요구 사항을 확인합니다. 적용 가능성, 규정 준수 기준 및 문제 해결 지침을 포함하여 전체 정의를 보려면 요구 사항을 선택합니다.

사용자 지정 팩 생성

사용자 지정 팩을 생성하여 조직에 특정한 보안 요구 사항을 그룹화합니다. 팩을 생성한 후 수동으로 팩에 요구 사항을 추가하거나, AWS관리형 요구 사항을 사용자 지정하거나, 문서를 업로드하여 요구 사항을 생성합니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택합니다.
4. 보안 요구 사항 팩 생성을 선택합니다.
5. 보안 요구 사항 팩 이름과 선택적 설명을 입력합니다.
6. 보안 요구 사항 팩 생성을 선택합니다.

> **Note**
>
> 팩을 생성한 후 소스 문서를 추가하거나 업로드하여 팩의 보안 요구 사항을 생성할 수 있습니다.

수동으로 보안 요구 사항 추가

사용자 지정 팩에 보안 요구 사항을 추가하여 AWS Security Agent가 적용하는 보안 표준을 정의합니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택한 다음 요구 사항을 추가할 팩을 선택합니다.
4. 수동으로 요구 사항 추가를 선택합니다.
5. 다음 필드를 구성합니다.
   a. 보안 요구 사항 이름 - 보안 제어를 식별하는 설명 이름을 입력합니다`enforce-encryption-at-rest`(예: 최대 80자).
   b. 설명 -이 보안 요구 사항이 확인하는 항목에 대한 간략한 설명을 제공합니다(최대 500자).
   c. 적용 가능성 -이 보안 요구 사항을 평가해야 하는 시나리오, 시스템 유형 또는 조건을 설명합니다. 요구 사항을 NOT_APPLICABLE(최대 10,000자)로 표시해야 하는 시나리오를 포함합니다.

d. 규정 준수 기준 -이 보안 요구 사항에 대한 규정 준수와 규정 미준수를 구성하는 요소를 정의합니다. AWS Security Agent가 규정 준수를 평가할 때 찾고 있는 지표, 예제 및 기술 세부 정보(최대 10,000자)를 제공합니다.

e. 문제 해결 지침(선택 사항) - 조직의 내부 문서 또는 표준 링크(최대 10,000자)를 포함하여 위반을 수정하는 방법에 대한 지침을 제공합니다.

6. 다음 중 하나를 수행하세요.

a. 요구 사항을 활성화하지 않고 생성하려면 보안 요구 사항 생성을 선택합니다.

b. 요구 사항을 생성하고 활성화하려면 보안 요구 사항 생성 및 활성화를 선택합니다.

> **Note**
>
> 요구 사항은 해당 요구 사항이 포함된 팩이 활성화된 경우에만 보안 검토 중에 평가됩니다. 팩 평가 여부를 제어하려면 팩을 활성화하거나 비활성화합니다.

AWS관리형 보안 요구 사항 사용자 지정

AWS관리형 요구 사항을 시작점으로 사용하는 사용자 지정 보안 요구 사항을 생성합니다. AWS Security Agent는 관리형 요구 사항의 요구 사항 세부 정보를 미리 채우고 조직에 맞게 편집합니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택한 다음 요구 사항을 추가할 사용자 지정 팩을 선택합니다.
4. 사용자 지정 보안 요구 사항 생성을 선택합니다.
5. 양식을 미리 채우려면 AWS관리형 보안 요구 사항 사용자 지정 섹션에서 템플릿으로 사용할 AWS 관리형 요구 사항을 검색하고 선택합니다.
6. 조직의 요구 사항에 맞게 필드를 편집합니다.
7. 다음 중 하나를 수행하세요.

a. 요구 사항을 활성화하지 않고 생성하려면 보안 요구 사항 생성을 선택합니다.

b. 요구 사항을 생성하고 즉시 활성화하려면 보안 요구 사항 생성 및 활성화를 선택합니다.

> **Note**
>
> AWS관리형 요구 사항을 사용자 지정하면 독립적인 사용자 지정 보안 요구 사항이 생성됩니다. AWS관리형 요구 사항을 나중에 변경해도 사용자 지정 버전에는 영향을 주지 않습니다.

문서를 업로드하여 요구 사항 생성

각 요구 사항을 직접 작성하는 대신 기존 보안 설명서에서 사용자 지정 팩에 대한 보안 요구 사항을 생성합니다. AWS Security Agent는 업로드한 문서를 읽고 보안 관련 콘텐츠를 식별하며 검토 및 편집할 수 있는 구조화된 요구 사항을 생성합니다. 전체 절차는 문서에서 보안 요구 사항 생성을 참조하세요. 업로드하는 문서에 포함할 항목에 대한 지침은 요구 사항 생성을 위한 문서 준비를 참조하세요.

> **Important**
>
> 새 소스 문서를 업로드하면 팩에 대한 모든 요구 사항이 다시 생성됩니다. 수동으로 추가한 요구 사항을 포함하여 기존 요구 사항이 모두 대체됩니다.

사용자 지정 보안 요구 사항 편집

사용자 지정 보안 요구 사항을 수정하여 정의, 기준 또는 수정 지침을 업데이트합니다. AWS관리형 팩에서는 요구 사항을 편집할 수 없습니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택한 다음 요구 사항이 포함된 팩을 선택합니다.
4. 편집하려는 요구 사항을 선택한 다음 사용자 지정 보안 요구 사항 편집을 선택합니다.
5. 필요에 따라 요구 사항 필드를 업데이트합니다. 보안 요구 사항 이름은 생성 후 변경할 수 없습니다.
6. 보안 요구 사항 업데이트를 선택합니다.

> **Note**
>
> 보안 요구 사항에 대한 변경 사항은 새 설계 검토 및 코드 검토에 적용됩니다. 기존 검토는 영향을 받지 않습니다.

팩 활성화 또는 비활성화

보안 요구 사항 팩을 활성화 또는 비활성화하여 AWS Security Agent가 포함된 요구 사항을 평가하는지 여부를 제어합니다. 팩을 단위로 활성화 및 비활성화합니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 팩 유형의 탭을 선택한 다음 팩을 선택합니다.
4. 다음 중 하나를 수행하세요.
   a. 팩을 활성화하려면 팩 활성화를 선택합니다.
   b. 팩을 비활성화하려면 팩 비활성화를 선택합니다.

> **Note**
>
> 팩이 활성화되면 보안 검토 중에 팩의 요구 사항이 평가됩니다. 팩이 비활성화되면 해당 요구 사항이 평가되지 않습니다.

사용자 지정 보안 요구 사항 삭제

AWS Security Agent에서 더 이상 평가하지 않도록 하려면 사용자 지정 보안 요구 사항을 삭제합니다.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택한 다음 요구 사항이 포함된 팩을 선택합니다.
4. 보안 요구 사항을 하나 이상 선택한 다음 보안 요구 사항 삭제를 선택합니다.
5. 삭제를 확인합니다.

> **Note**
>
> 보안 요구 사항을 삭제하면 향후 검토에서 더 이상 평가되지 않습니다. 요구 사항은 과거 검토에서 읽기 전용 결과로 계속 표시됩니다.

보안 요구 사항 정의 모범 사례

보안 요구 사항을 생성할 때 다음 지침에 따라 AWS Security Agent가 애플리케이션을 정확하게 평가하고 실행 가능한 결과를 제공하는 데 도움이 됩니다.

보안 요구 사항 이름 - 보안 제어를 식별하는 명확하고 구체적인 이름을 사용합니다. 요구 사항의 목적을 전달하지 않는 일반 용어는 사용하지 마세요.

설명 - 제어 검사의 내용과 이를 통해 완화되는 위험을 설명합니다. 이를 통해 사용자는 요구 사항이 중요한 이유를 이해할 수 있습니다.

적용성 - 요구 사항을 평가해야 하는 워크로드, 시스템 유형 또는 조건을 설명합니다. 거짓 긍정을 방지하기 위해 특정 시나리오에서 요구 사항을 NOT_APPLICABLE로 표시해야 하는 시기를 명시합니다. "이 제어는 " 및 "not_APPLICABLE로 표시... " 명확한 범위 경계를 설정하는 모든 워크로드에 적용됩니다.

규정 준수 기준 - 규정 준수를 보여주는 부분과 규정 미준수를 나타내는 부분의 두 부분으로 구성됩니다. 기술 지표를 구체적으로 설명하고 엣지 케이스를 포함합니다. 위반을 나타내는 패턴을 사용하여 "설계는 규정을 준수합니다." 다음에 기술 세부 정보가 옵니다. "설계는 규정을 준수하지 않습니다."로 시작합니다.

해결 지침 - 기술 세부 정보 및 구성 예제에 대한 지침을 제공합니다. 개발자가 위반을 해결하는 데 필요한 리소스를 확보할 수 있도록 조직의 내부 설명서에 대한 링크를 포함합니다.

다음 단계

보안 요구 사항 팩을 구성한 후:

- 설계 검토 또는 코드 검토를 생성하여 활성화한 팩을 기준으로 애플리케이션을 평가합니다.
- 침투 테스트를 활성화하여 설계 및 코드 검토를 보완합니다.
- 사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한을 부여합니다.

AWS 관리형 보안 요구 사항 팩

AWS 관리형 보안 요구 사항 팩은 업계 표준 및 모범 사례에 따라가 AWS 생성하고 유지 관리하는 보안 요구 사항의 모음입니다. 관리형 팩을 사용하면 요구 사항을 직접 작성하지 않고도 설계 검토 및 코드 검토 중에 요구 사항을 기준으로 애플리케이션을 평가할 수 있습니다.

관리형 팩의 보안 요구 사항은 읽기 전용입니다. 변경할 수 없습니다. AWS관리형 요구 사항을 템플릿으로 사용하여 사용자 지정 요구 사항을 생성한 다음 조직에 맞게 사본을 편집할 수 있습니다. 자세한 내용은 보안 요구 사항 관리를 참조하세요.

AWS 는 시간이 지남에 따라 관리형 팩을 업데이트합니다. 가 관리형 팩의 요구 사항을 AWS 추가하거나 수정하면 팩이 활성화된 모든 계정에 변경 사항이 적용됩니다.

> **Note**
>
> 규정 준수 프레임워크 팩은 특정 프레임워크 규정 준수와 관련이 있을 수 있는 보안 요구 사항을 식별하는 데 도움이 되도록 설계되었습니다. 규정 준수를 평가하거나 감사를 통과할 것이라고 보장하지 않습니다.

AWS 관리형 팩: ASA 기본 팩

대부분의 워크로드에 적용되는 보안 요구 사항의 기준 세트입니다. 이 팩은 인증 및 권한 부여, 데이터 보호, 로깅 및 입력 검증과 같은 일반적인 애플리케이션 보안 문제를 다룹니다. 이 팩을 활성화하여 설계 및 코드 검토에 대한 보안 기준을 설정합니다.

AWS 관리형 팩: AWS Well-Architected Pack

AWS Well-Architected Framework의 보안 원칙에서 파생된 보안 요구 사항입니다. 이 팩은 데이터 보호, 자격 증명 및 권한 관리, 보안 이벤트 감지 및 대응을 위한 AWS 지침에 따라 애플리케이션을 평가합니다. 프레임워크에 대한 자세한 내용은 보안 원칙 - AWS Well-Architected Framework를 참조하세요.

AWS 관리형 팩: NIST CSF Pack

NIST 사이버 보안 프레임워크(CSF)에서 파생된 보안 요구 사항입니다. 이 팩은 자격 증명 및 액세스 관리, 데이터 보안, 보호 기술과 같이 프레임워크에서 가져온 제어 영역을 기준으로 애플리케이션을 평가합니다. 이 팩을 활성화하여 설계 및 코드 검토를 NIST CSF와 일치시킵니다.

AWS 관리형 팩: PCI DSS 팩

PCI DSS(지불 카드 산업 데이터 보안 표준)에서 파생된 보안 요구 사항입니다. 이 팩은 액세스 제어, 전송 중 및 저장 데이터 암호화, 로깅과 같은 카드 소지자 데이터 처리와 관련된 제어 영역에 대해 애플리케이션을 평가합니다. 이 팩을 활성화하여 설계 및 코드 검토를 PCI DSS와 일치시킵니다.

문서에서 보안 요구 사항 생성

기존 보안 설명서를 업로드하여 사용자 지정 팩의 보안 요구 사항을 생성합니다. AWS Security Agent는 업로드한 문서를 읽고, 보안 관련 콘텐츠를 식별하고, 적용 가능성, 규정 준수 기준 및 수정 지침을 사용하여 구조화된 보안 요구 사항을 생성합니다. 생성된 요구 사항을 검토에 사용하기 전에 검토하고 편집할 수 있습니다.

보안 정책, 표준 또는 지침이 이미 문서화된 경우 각 요구 사항을 직접 작성하는 대신 이를 사용합니다. 업로드하는 문서에 포함할 항목에 대한 지침은 요구 사항 생성을 위한 문서 준비를 참조하세요.

지원되는 파일 형식

다음 파일 형식을 업로드할 수 있습니다.

- DOC
- DOCX
- MD
- PDF
- TXT

요구 사항 생성

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택합니다.
4. 팩을 생성하거나 요구 사항을 생성할 기존 사용자 지정 팩을 선택합니다.
5. 파일 선택을 선택하거나 문서를 업로드 영역으로 끌어서 놓습니다.
6. 요구 사항 생성을 선택합니다.
7. AWS Security Agent가 문서를 처리할 때까지 기다립니다. 생성은 백그라운드에서 실행되며 대용량 파일의 경우 몇 분 정도 걸릴 수 있습니다. 생성이 진행되는 동안에는 팩에 대해 다른 업로드를 시작할 수 없습니다.
8. 생성된 보안 요구 사항을 검토합니다. 필요에 따라 요구 사항을 편집, 삭제 또는 추가합니다. AWS Security Agent가 보안 검토 중에 요구 사항을 평가하도록 하려면 팩을 활성화합니다.

> **Note**
>
> AWS Security Agent는 워크로드 수준에서 요구 사항을 생성하고 문서의 보안 관련 콘텐츠에 중점을 둡니다. 생성하는 요구 사항의 수는 사용자가 제공하는 콘텐츠에 따라 달라집니다. 생성된 요구 사항을 검토하여 의도를 반영하는지 확인합니다.

요구 사항을 새 업로드로 바꾸기

새 소스 문서를 업로드하면 팩에 대한 요구 사항이 다시 생성됩니다.

> **Important**
>
> 새 소스 문서를 팩에 업로드하면 AWS Security Agent는 해당 팩에 대한 모든 요구 사항을 다시 생성합니다. 수동으로 추가한 요구 사항을 포함하여 기존 요구 사항이 모두 대체됩니다. 현재 요구 사항을 유지하려면 새 문서를 업로드하지 마십시오.

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 탐색 창에서 보안 요구 사항을 선택합니다.
3. 사용자 지정 보안 요구 사항 팩 탭을 선택한 다음 팩을 선택합니다.
4. 새 업로드를 시작하고 새 소스 문서를 업로드하면 기존 요구 사항이 대체됨을 확인합니다.
5. 파일을 선택한 다음 요구 사항 생성을 선택합니다.

다음 단계

- 생성된 요구 사항을 검토하고 활성화합니다. 보안 요구 사항 관리를 참조하세요.
- 설계 검토 또는 코드 검토를 생성하여 팩을 기준으로 애플리케이션을 평가합니다.

요구 사항 생성을 위한 문서 준비

문서에서 보안 요구 사항을 생성하면 AWS Security Agent는 문서를 읽고 찾은 보안 관련 콘텐츠에서 구조화된 요구 사항을 생성합니다. 특정 방식으로 문서의 형식을 지정하거나 적용 가능성, 규정 준수 기준 및 수정 지침을 미리 작성할 필요가 없습니다. AWS Security Agent는 사용자가 제공하는 콘텐츠에서 문서를 추출합니다. 이미 가지고 있는 자체 단어로 작성된 보안 설명서를 업로드합니다.

이 페이지에서는 AWS Security Agent가 정확하고 유용한 요구 사항을 생성하도록 포함할 내용을 설명합니다. 업로드 절차 및 파일 제한은 문서에서 보안 요구 사항 생성을 참조하세요.

포함할 항목

AWS Security Agent는 보안 기대치를 설명하는 콘텐츠를 찾습니다. 다음 주제를 다루는 문서는 가장 강력한 요구 사항을 생성합니다.

- 액세스 제어 및 권한 부여
- Authentication
- 데이터 보호 및 암호화
- 네트워크 보안
- 로깅 및 감사
- 인시던트 대응
- 취약성 및 패치 관리
- 워크로드에 적용되는 규정 준수 의무

좋은 소스 문서에는 보안 정책, 엔지니어링 표준, 제어 카탈로그, 아키텍처 지침 및 보안 코딩 표준이 포함됩니다.

워크로드 수준에서 쓰기

AWS Security Agent는 설계 및 코드 검토 중에 평가하는 것과 동일한 범위인 개별 워크로드 또는 애플리케이션에 적용되는 요구 사항을 생성합니다. 보안 워크로드를 구축하고 운영하는 방법을 설명하는 콘텐츠는 요구 사항을 생성합니다. 다중 계정 전략, 루트 사용자 관리, 계정 수준 로깅 설정과 같은 조직 전체의 거버넌스 주제는이 범위를 벗어나며 워크로드 요구 사항을 생성하지 않습니다.

단일 구현이 아닌 결과 설명

규정된 단일 제품 또는 구성이 아닌 예상한 보안 결과를 설명합니다. AWS Security Agent는 필요한 보안 기능에 초점을 맞추고 둘 이상의 유효한 구현을 허용하는 요구 사항을 생성합니다. 예를 들어 "저장 데이터가 암호화됨"은 하나의 특정 서비스 또는 설정을 지정하는 문보다 더 명확하고 광범위하게 적용할 수 있는 요구 사항을 생성합니다.

좋은 모습에 대해 구체적으로 설명하세요.

문서가 예상 동작을 보다 구체적으로 설명할수록 AWS Security Agent가 규정 준수 기준을 더 잘 정의할 수 있습니다. 가능한 경우 규정 준수 설계가 보여주는 것과 위반을 나타내는 것을 설명합니다. 모호하거나 순전히 욕구적인 문은 구체적인 워크로드 관련 문보다 요구 사항이 더 적고 약합니다.

반환되는 내용 검토

생성된 각 요구 사항에는 AWS Security Agent가 문서에서 도출한 이름, 적용 가능성, 규정 준수 기준 및 수정 지침이 포함됩니다. 생성된 요구 사항을 검토하고, 세분화가 필요한 사항을 편집하고, AWS Security Agent가 평가할 요구 사항을 활성화합니다. 자세한 내용은 보안 요구 사항 관리를 참조하세요.

사용자 및 액세스 관리

각 에이전트 스페이스에 액세스할 수 있는 사용자를 제어하고 AWS Security Agent가 사용하는 IAM 역할 및 암호화 키를 구성합니다.

사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한 부여

AWS Security Agent는 설정 중에 에이전트 공간을 구성한 방법에 따라 사용자가 웹 애플리케이션에 액세스할 수 있는 두 가지 방법을 제공합니다.

액세스 방법 개요

IAM Identity Center(SSO) - 에이전트 스페이스를 생성할 때 IAM Identity Center를 활성화한 경우 사용자는 SSO를 통해 직접 웹 애플리케이션에 액세스할 수 있습니다. AWS Security Agent 콘솔을 통해 Agent Space에 사용자를 할당하고 사용자는 Identity Center 자격 증명을 사용하여 로그인합니다.

관리자 액세스 - AWS 콘솔 액세스 권한이 있는 사용자는 AWS 관리 콘솔의 에이전트 스페이스 개요 페이지에 있는 관리자 액세스 링크를 통해 웹 애플리케이션을 시작할 수 있습니다.

IAM Identity Center(SSO)를 사용하여 액세스 권한 부여

IAM Identity Center로 에이전트 스페이스를 구성한 경우 AWS Security Agent 콘솔을 사용하여 에이전트 스페이스에 사용자를 할당할 수 있습니다.

AWS Security Agent 콘솔을 통해 사용자 할당

1. AWS 보안 에이전트 관리 콘솔에서 에이전트 공간으로 이동합니다.

2. 웹 앱 탭을 선택합니다.
3. 사용자 테이블에서 사용자 추가를 선택합니다.
4. IAM Identity Center에서 기존 사용자를 선택하거나 새 사용자를 생성합니다.
5. 사용자 할당을 확인합니다.

> **Tip**
>
> 에이전트 스페이스에 할당된 사용자는 SSO 자격 증명으로 IAM Identity Center를 통해 로그인하여 웹 애플리케이션에 액세스할 수 있습니다.

SSO를 사용하여 웹 애플리케이션에 액세스

사용자가 에이전트 스페이스에 할당된 후:

1. 사용자는 에이전트 스페이스의 웹 애플리케이션 URL로 이동합니다.

> **Tip**
>
> 웹 앱 URL 복사를 선택하여 AWS Security Agent 콘솔의 에이전트 스페이스 세부 정보 페이지에서 웹 앱 URL을 찾습니다. 사용자는 쉽게 액세스할 수 있도록이 URL을 북마크해야 합니다.

2. 사용자는 SSO 자격 증명을 사용하여 로그인합니다.
3. 인증 후 사용자는 에이전트 공간을 선택하고 보안 평가를 시작할 수 있습니다.

IAM 전용 액세스 권한 부여

IAM 전용 액세스 권한으로 에이전트 스페이스를 구성한 경우 AWS 콘솔 액세스 권한이 있는 사용자는 관리자 액세스 링크를 통해 웹 애플리케이션을 시작할 수 있습니다.

관리자 액세스 링크 사용

1. AWS Security Agent 콘솔에 로그인합니다.
2. 액세스하려는 에이전트 공간으로 이동합니다.
3. 에이전트 스페이스 랜딩 페이지의 웹 앱 탭에서 관리자 액세스 버튼을 선택하여 웹 애플리케이션을 시작합니다.

4. 웹 애플리케이션이 새 탭에서 열리고 사용자가 자동으로 인증됩니다.

> **Note**
>
> 관리자 액세스 링크는 적절한 AWS Security Agent 권한이 있는 AWS 콘솔에 이미 인증된 사용자만 사용할 수 있습니다. 이 메서드에는 IAM Identity Center 구성이 필요하지 않습니다.

다음 단계

사용자에게 웹 애플리케이션에 대한 액세스 권한을 부여한 후:

- 사용자는 침투 테스트 구성 및 실행을 생성하고 관리할 수 있습니다.
- 사용자는 설계 검토를 생성하고 관리할 수 있습니다.
- 사용자는 보안 조사 결과 및 수정 지침을 볼 수 있습니다.
- 보안 조사 결과에 대한 알림 기본 설정 구성

AWS Security Agent에 대한 IAM 역할 생성

AWS Security Agent는 다음 세 가지 방법으로 IAM 역할을 사용합니다.

1. 애플리케이션 역할: AWS Security Agent 애플리케이션을 생성할 때 사용됩니다. IAM Identity Center 및 관리자 액세스 링크 사용 사례의 경우 서비스는이 역할을 수임하여 WebApp 사용자에게 AWS Security Agent APIs와 상호 작용할 수 있는 권한을 부여합니다.
2. 침투 테스트 서비스 역할: 에이전트 공간을 생성할 때 사용 가능한 역할 목록으로 지정됩니다. 나중에 WebApp 사용자는 침투 테스트를 생성할 때 이러한 역할 중 하나를 선택합니다. AWS Security Agent 서비스는이 역할을 수임하여 테스트 중에 AWS 리소스에 액세스합니다.
3. 액터 역할: 대상 웹 애플리케이션(예: AWS API Gateway APIs)에 대한 요청을 인증하고 권한을 부여하는 데 사용됩니다. 이러한 역할은 에이전트 공간 생성 중에 제공됩니다. AWS Security Agent 에이전트는 대상 애플리케이션과 상호 작용하기 위해 액터 역할을 수임합니다.

애플리케이션 역할

Application Role은 서비스에서 AWS Security Agent 애플리케이션을 생성할 때 사용됩니다. IAM Identity Center 및 관리자 액세스 링크 인증 시나리오의 경우 AWS Security Agent 서비스는이 역할을

수임하여 WebApp 사용자에게 AWS Security Agent APIs와 상호 작용하는 데 필요한 권한을 부여합니다.

필수 권한

이 역할에는 다음과 같은 권한이 필요합니다.

- AWS Security Agent API 작업 호출
- 애플리케이션 구성 데이터 읽기 및 쓰기
- 사용자 세션 정보 액세스
- WebApp 사용자의 인증 토큰 관리

신뢰 정책

신뢰 정책은 AWS Security Agent 서비스가이 역할을 수임하도록 허용해야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

권한 정책

역할에는 다음에 대한 권한이 포함되어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "securityagent:GetApplication",
        "securityagent:UpdateApplication",
```

```
        "securityagent:ListAgentInstances",
        "securityagent:CreatePentestSession"
      ],
      "Resource": "arn:aws:securityagent:*:*:application/*"
    }
  ]
}
```

> **Note**
>
> 특정 애플리케이션 요구 사항과 최소 권한 원칙에 따라 권한을 사용자 지정합니다.

침투 테스트 서비스 역할

침투 테스트 서비스 역할은 에이전트 공간을 사용 가능한 역할 목록으로 생성할 때 지정됩니다. WebApp 사용자는 침투 테스트를 생성할 때 이러한 역할 중 하나를 선택합니다. 그런 다음 AWS Security Agent 서비스는이 역할을 수임하여 AWS 리소스에 액세스하고 테스트합니다.

필수 권한

이 역할에는 침투 테스트 중에 AWS 리소스에 액세스하고 분석할 수 있는 권한이 필요합니다.

- VPC 구성 및 네트워크 토폴로지 읽기 및 설명
- EC2 인스턴스, 보안 그룹 및 네트워크 ACLs 검사
- IAM 정책 및 리소스 권한 분석
- CloudWatch 로그 및 지표 읽기
- 보안 테스트와 관련된 AWS 서비스 구성에 액세스

신뢰 정책

신뢰 정책은 AWS Security Agent 서비스가 침투 테스트 작업을 위해이 역할을 수임하도록 허용해야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "your-external-id"
        }
      }
    }
  ]
}
```

> **Note**
>
> 교차 계정 또는 서비스 액세스를 허용할 때 추가 보안을 위해 외부 ID를 사용합니다.

권한 정책

역할에는 AWS 리소스에 대한 읽기 전용 액세스가 포함되어야 합니다. 다음 관리형 정책을 사용하는 것이 좋습니다.

- `SecurityAudit` - 보안 감사를 위한 AWS 관리형 정책
- `ViewOnlyAccess` - 대부분의 AWS 서비스에 대한 읽기 전용 액세스

또는 특정 권한이 있는 사용자 지정 정책을 생성합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "vpc:Describe*",
        "iam:Get*",
        "iam:List*",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "cloudwatch:Describe*",
```

```
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Important**
>
> 침투 테스트 범위에 필요한 최소 권한만 부여합니다. 보안 테스트에 포함할 AWS 서비스에 따라 권한을 검토하고 조정합니다.

액터 역할

액터 역할은 침투 테스트 중에 대상 웹 애플리케이션에 대한 요청을 인증하고 승인하는 데 사용됩니다. 이러한 역할은 에이전트 공간 생성 중에 제공되며 AWS Security Agent 에이전트는 이러한 역할이 대상 애플리케이션 엔드포인트(예: AWS API Gateway APIs, Lambda 함수 URLs 또는 기타 AWS 호스팅 애플리케이션)와 상호 작용한다고 가정합니다.

필수 권한

이 역할에는 다음과 같은 권한이 필요합니다.

- API Gateway 엔드포인트 호출
- Lambda 함수 실행
- 애플리케이션별 AWS 리소스에 액세스
- 대상 애플리케이션의 인증 메커니즘으로 인증
- 애플리케이션 엔드포인트에 대해 HTTP 작업 수행

신뢰 정책

신뢰 정책은 AWS Security Agent 에이전트 서비스가이 역할을 수임하도록 허용해야 합니다.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "your-external-id"
        }
      }
    }
  ]
}
```

권한 정책

권한은 대상 애플리케이션 아키텍처에 따라 다릅니다. 다음은 일반적인 시나리오의 예입니다.

API Gateway 애플리케이션의 경우

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "execute-api:Invoke"
      ],
      "Resource": "arn:aws:execute-api:us-east-1:*:your-api-id/*"
    }
  ]
}
```

Lambda 함수 URLs 경우

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "lambda:InvokeFunctionUrl",
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:us-east-1:*:function:your-function-name"
    }
  ]
}
```

Cognito 인증을 사용하는 Application Load Balancer의 경우

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-idp:InitiateAuth",
        "cognito-idp:RespondToAuthChallenge"
      ],
      "Resource": "arn:aws:cognito-idp:us-east-1:*:userpool/your-user-pool-id"
    }
  ]
}
```

> **Important**
>
> 대상 애플리케이션의 인증 및 권한 부여 요구 사항에 맞게 액터 역할 권한을 구성합니다. 역할은 Security Agent가 침투 테스트 중에 시뮬레이션할 사용자 또는 서비스와 동일한 수준의 액세스 권한을 가져야 합니다.

AWS Security Agent에 사용되는 고객 관리형 키

기본적으로 AWS Security Agent는 AWS관리형 암호화 키를 사용하여 저장된 모든 데이터를 암호화합니다. 선택적으로 AWS Key Management Service(AWS KMS)의 고객 관리형 키를 사용하여 데이터를 암호화하여 리소스를 보호하는 암호화 키를 완벽하게 제어할 수 있습니다.

AWS Security Agent는 리소스 수준 고객 관리형 키를 지원합니다. 에이전트 스페이스 또는 통합과 같은 최상위 리소스를 생성할 때 KMS 키를 지정하여 해당 리소스 및 해당 하위 리소스에 속하는 모든 데이터를 암호화할 수 있습니다. 예를 들어 에이전트 스페이스를 생성할 때 고객 관리형 키를 지정하

면 에이전트 스페이스 구성, 침투 테스트 구성, 작업 및 실행 세부 정보, 보안 조사 결과, 검색된 엔드포인트 및 스크린샷을 포함하여 해당 에이전트 스페이스와 관련된 모든 데이터가 암호화됩니다. S3 버킷, CloudWatch Logs 로그 그룹 또는 Secrets Manager 보안 암호와 같이 자체 고객 관리형 키로 이미 암호화된 AWS 리소스를 제공할 수도 있습니다. 필요한 KMS 권한은 섹션을 참조하세요the section called “필수 KMS 권한”.

고객 관리형 키의 작동 방식

AWS Security Agent는 AWS KMS 계층적 키링을 사용하여 고객 관리형 키로 데이터를 암호화합니다. 리소스 생성 중에 KMS 키를 지정하면 서비스는 KMS 키로 보호되는 브랜치 키를 생성하여 Amazon DynamoDB 기반 키 스토어에 저장합니다. 각 암호화 작업에 대해 계층적 키링은 각 요청에 대한 고유한 데이터 암호화 키를 암호화하는 활성 브랜치 키에서 고유한 래핑 키를 도출합니다.

계층적 키링은 다음과 같은 이점을 제공합니다.

- 리소스별 암호화 범위 - 각 최상위 리소스에는 자체 브랜치 키가 있으므로 서로 다른 리소스에 대한 데이터는 별도의 키로 암호화됩니다.
- 자동 키 교체 - AWS Security Agent는 브랜치 키를 주기적으로 교체합니다. 교체 후 새 데이터는 새 브랜치 키 버전으로 암호화되고 이전 버전은 이전에 암호화된 데이터를 해독하기 위해 유지됩니다.

암호화 컨텍스트

AWS Security Agent는 모든 암호화 작업에서 KMS 키를 사용하여 암호화 컨텍스트를 사용합니다. 암호화 컨텍스트는 암호화 작업에 대해 추가로 인증된 데이터를 제공하는 비밀이 아닌 키-값 페어 세트입니다.

암호화 컨텍스트 키는 형식을 따릅니다. `aws:securityagent:_<resource-type>_`여기서 *`<resource-type>`* 는 암호화되는 리소스 유형(예: `agent-space` 또는 `integration`)입니다. 값은 리소스의 Amazon 리소스 이름(ARN)입니다.

> **Note**
>
> 통합의 경우 암호화 컨텍스트 키에 `aws-crypto-ec:` 접두사가 포함되어 형식이 됩니다`aws-crypto-ec:aws:securityagent:integration`.

암호화 컨텍스트를 사용하여 KMS 키 정책의 범위를 특정 리소스 유형으로 좁힐 수 있습니다. 자세한 내용은 the section called “필수 KMS 권한” 단원을 참조하십시오.

기본 암호화

고객 관리형 키를 지정하지 않고 리소스를 생성하면 AWS Security Agent는 기본 스토리지 서비스(Amazon DynamoDB 및 Amazon S3)에서 제공하는 AWS관리형 암호화 키를 사용하여 리소스를 암호화합니다. 기본 암호화에는 추가 구성이 필요하지 않습니다.

애플리케이션의 기본 KMS 키

애플리케이션 수준에서 기본 KMS 키를 설정할 수 있습니다. 기본 KMS 키가 구성된 경우 리소스 생성 중에 KMS 키를 명시적으로 지정하지 않으면 AWS Security Agent는 이를 새 에이전트 스페이스 및 통합에 대한 대체 키로 사용합니다.

기본 KMS 키를 설정하려면 AWS CLI 또는 SDK를 사용하여 애플리케이션을 생성하거나 업데이트할 때 `defaultKmsKeyId` 파라미터를 지정합니다. 콘솔에 애플리케이션 설정 중에 기본 KMS 키를 지정하라는 메시지가 표시됩니다.

리소스 생성 중에 KMS 키를 명시적으로 지정하면 애플리케이션 수준 기본값보다 우선합니다.

사전 조건

고객 관리형 키를 구성하기 전에 다음 사전 조건을 완료합니다.

- KMS에서 대칭 암호화 AWS KMS 키를 생성합니다. 키는 다음 요구 사항을 충족해야 합니다.
  - 키 유형: 대칭
  - 키 사용: 암호화 및 복호화
  - 키 사양: `SYMMETRIC_DEFAULT`
  - 키 상태: 활성화됨

    지침은 Key Management Service 개발자 안내서의 키 생성을 참조하세요. AWS
- AWS Security Agent에 필요한 권한을 부여하도록 KMS 키 정책 및 IAM 정책을 구성합니다. 자세한 내용은 the section called “필수 KMS 권한”을 참조하세요.

필수 KMS 권한

AWS Security Agent에는 두 가지 시나리오에서 KMS 권한이 필요합니다.

- AWS Security Agent 내에서 데이터 암호화 - 에이전트 스페이스 또는 통합에 대한 고객 관리형 키를 지정하면 서비스에서 데이터에 대한 암호화 작업에 해당 키를 사용할 수 있는 권한이 필요합니다.

- CMK로 암호화된 AWS 리소스 사용 - 고객 관리형 키(예: S3 버킷, CloudWatch Logs 로그 그룹 또는 Secrets Manager 보안 암호)로 암호화된 AWS 리소스를 제공하는 경우 서비스는 이러한 키를 사용하여 암호화된 리소스에 액세스할 수 있는 권한이 필요합니다.

AWS Security Agent는 작업에 따라 다른 자격 증명을 사용하여 KMS 키에 액세스합니다.

- AWS 관리 콘솔 작업 - 관리자가 AWS 관리 콘솔에서 리소스를 생성하거나 관리할 때(예: 에이전트 공간 생성 또는 애플리케이션 업데이트) 서비스는 관리자 역할을 사용하여 AWS KMS를 호출합니다.
- 웹 애플리케이션 작업 - IAM Identity Center 사용자가 AWS Security Agent 웹 애플리케이션을 통해 데이터에 액세스하는 경우(예: 침투 테스트 결과 보기 또는 침투 테스트 시작) 서비스는 AWS Security Agent 설정 중에 생성된 애플리케이션 역할을 사용하여 AWS KMS를 호출합니다.
- 고객 제공 리소스 액세스 - 서비스가 침투 테스트 중에 고객 제공 AWS 리소스(예: S3 버킷, CloudWatch Logs 로그 그룹 및 Secrets Manager 보안 암호)에 액세스할 때 침투 테스트 서비스 역할을 사용하여 AWS KMS를 호출합니다.
- 비동기 워크플로 - 사용자 세션 외부에서 실행되는 백그라운드 작업(예: 침투 테스트 실행, 코드 검토 처리 및 브랜치 키 교체)의 경우 서비스는 자체 서비스 보안 주체(`securityagent.amazonaws.com`)를 사용하여 사용자를 대신하여 AWS KMS를 호출합니다.

다음 섹션에서는 각 자격 증명에 필요한 권한을 설명합니다.

키 정책

KMS 키 정책은 암호화 작업에 키를 사용할 수 있는 권한을 AWS Security Agent에 부여해야 합니다. 필수 키 정책 설명은 암호화하려는 리소스 유형과 서비스에 제공하는 CMK 암호화 AWS 리소스에 따라 달라집니다.

에이전트 스페이스에 대한 키 정책

다음 키 정책은 AWS Security Agent에 침투 테스트 결과 및 스크린샷을 포함하여 에이전트 스페이스 데이터를 암호화하고 해독하는 데 필요한 권한을 부여합니다.

정책에서 다음 자리 표시자 값을 바꿉니다.

- *`111122223333`* - AWS 계정 ID
- *`MyRole`* - 콘솔에서 AWS Security Agent를 관리하는 데 사용하는 IAM 역할
- *`MyApplicationRole`* - AWS Security Agent 설정 중에 생성된 애플리케이션 역할
- *`us-east-1`* - AWS Security Agent를 사용하는 AWS 리전

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKeyMetadataValidationForApplicationAndAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowUseOfHierarchicalKeyringForAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
 "arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        },
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
```

```
      "Sid": "AllowSynchronousDataAccessForAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyApplicationRole"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
"arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        },
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowAsynchronousDataAccessForAgentSpaces",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
"arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        }
      }
    },
    {
```

```
      "Sid": "AllowAsynchronousDataAccessForCodeRemediation",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*"
    }
  ]
}
```

Important

브랜치 키를 교체하려면 KMS 키 정책이 AWS Security Agent 서비스 보안 주체(`securityagent.amazonaws.com`)에게 `kms:ReEncryptTo`, 및 `kms:ReEncryptFrom` 권한을 부여해야 합니다. 그렇지 않으면 `kms:GenerateDataKeyWithoutPlaintext`브랜치 키 교체가 실패합니다. 필요한 권한에 대한 자세한 내용은 브랜치 키 교체를 참조하세요.

통합을 위한 키 정책

다음 키 정책은 코드 검토 조사 결과를 포함하여 통합 데이터를 암호화하고 복호화하는 데 필요한 권한을 AWS Security Agent에 부여합니다.

정책에서 다음 자리 표시자 값을 바꿉니다.

- *111122223333* - AWS 계정 ID
- *MyRole* - 콘솔에서 AWS Security Agent를 관리하는 데 사용하는 IAM 역할
- *us-east-1* - AWS Security Agent를 사용하는 AWS 리전

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKeyAccessValidationForIntegrations",
      "Effect": "Allow",
      "Principal": {
```

```
          "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:securityagent:integration":
"arn:aws:securityagent:us-east-1:111122223333:integration/*"
        },
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowKeyMetadataValidationForIntegrations",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "AllowAsynchronousDataAccessForIntegrations",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "ArnLike": {
          "aws:SourceArn": "arn:aws:securityagent:us-east-1:111122223333:integration/*"
        },
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:securityagent:integration":
 "arn:aws:securityagent:us-east-1:111122223333:integration/*"
        }
      }
    }
  ]
}
```

> **Note**
>
> 여러 리소스 유형에 동일한 KMS 키를 사용하려는 경우 에이전트 스페이스, 통합 및 보안 요구 사항 팩 키 정책 설명을 단일 키 정책으로 결합할 수 있습니다.

보안 요구 사항 팩에 대한 키 정책

다음 키 정책은 보안 요구 사항 팩 데이터를 암호화하고 해독하는 데 필요한 권한을 AWS Security Agent에 부여합니다. 보안 요구 사항 팩은 웹 애플리케이션 역할을 사용하지 않습니다. 이 정책은 두 가지 액세스 경로를 사용합니다.

- 동기 경로 - API를 호출하면 서비스는 전달된 자격 증명을 사용하여 사용자를 대신하여(`kms:ViaService` 조건을 통해) AWS KMS를 호출합니다.
- 비동기 경로 - 팩을 사용할 수 있는 비동기 작업의 경우 서비스는 자체 서비스 보안 주체를 사용하여 AWS KMS를 직접 호출합니다.

정책에서 다음 값을 바꿉니다.

- *111122223333* - AWS 계정 ID
- *MyRole* - 보안 요구 사항 팩 작업을 호출하는 데 사용하는 IAM 역할
- *us-east-1* - AWS Security Agent를 사용하는 AWS 리전

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "AllowKeyMetadataValidation",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowUseOfHierarchicalKeyringForSecurityPacks",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyRole"
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:security-requirement-pack":
"arn:aws:securityagent:us-east-1:111122223333:security-requirement-pack/*"
        }
      }
    },
    {
      "Sid": "AllowAsynchronousDataAccessForSecurityPacks",
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
```

```
      },
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:security-requirement-pack":
 "arn:aws:securityagent:us-east-1:111122223333:security-requirement-pack/*"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:securityagent:us-east-1:111122223333:security-
requirement-pack/*"
        }
      }
    }
  ]
}
```

정책에는 다음 세 가지 명령문이 포함되어 있습니다.

- AllowKeyMetadataValidation - AWS Security Agent가 리소스 생성 중에 CMK를 지정할 때 고객 관리형 키가 존재하는지, 활성화되었는지, 대칭 암호화를 사용하는지 검증할 수 `kms:DescribeKey` 있도록 권한을 부여합니다. `kms:ViaService` 조건을 사용하여 호출이 서비스를 통해 시작되도록 합니다.
- AllowUseOfHierarchicalKeyringForSecurityPacks – 계층적 키링 작업에 대한 권한 부여`kms:GenerateDataKeyWithoutPlaintext`(새 브랜치 키 생성) `kms:ReEncryptTo` 및 `kms:ReEncryptFrom` (기존 브랜치 키 교체) 및 `kms:Decrypt` (기존 브랜치 키 검색) 이러한 작업은 동기 경로를 통해 전달된 자격 증명을 사용하며 암호화 컨텍스트 조건에 따라 보안 요구 사항 팩 리소스로 범위가 지정됩니다.
- AllowAsynchronousDataAccessForSecurityPacks - 호출자 자격 증명을 사용할 수 없는 경우 비동기 작업을 위해 AWS Security Agent 서비스 보안 주체`kms:ReEncryptFrom`에 `kms:GenerateDataKeyWithoutPlaintext``kms:Decrypt`, `kms:ReEncryptTo`, 및를 부여합니다.

에이전트 스페이스 키 정책과 달리 보안 요구 사항 팩 정책에는 웹 애플리케이션 역할 보안 주체가 포함되지 않습니다. 보안 요구 사항 팩은 AWS Security Agent 웹 애플리케이션이 아닌 관리자 역할을 사

용하여 AWS Management Console을 통해 액세스됩니다. 모든 동기 작업은 단일 호출자 역할(를 통해)을 사용합니다`kms:ViaService`.

> **Note**
>
> 에이전트 스페이스와 보안 요구 사항 팩 모두에 동일한 KMS 키를 사용하는 경우 두 섹션의 키 정책 설명을 단일 키 정책으로 결합할 수 있습니다.

CMK 암호화 AWS 리소스에 대한 키 정책

고객 관리형 키로 암호화된 AWS 리소스를 AWS Security Agent에 제공하는 경우 각 리소스의 KMS 키에 대한 키 정책을 업데이트하여 필요한 권한을 부여해야 합니다. 이는 다음 리소스에 적용됩니다.

- S3 버킷 - 고객 관리형 키(SSE-KMS)로 암호화된 S3 버킷에서 학습 리소스를 제공하는 경우 키 정책은 침투 테스트 서비스 역할이 객체를 복호화하도록 허용해야 합니다.
- Secrets Manager 보안 암호 - 침투 테스트 보안 인증 정보가 고객 관리형 키로 암호화된 Secrets Manager 보안 암호에 저장되는 경우 키 정책은 침투 테스트 서비스 역할이 해당 보안 암호를 해독하도록 허용해야 합니다. 웹 애플리케이션이 사용자를 대신하여 보안 암호를 생성하는 경우 키 정책은 애플리케이션 역할이 해당 보안 암호를 암호화하도록 허용해야 합니다.
- CloudWatch Logs 로그 그룹 - 침투 테스트 실행 로그를 저장하는 데 사용되는 CloudWatch Logs 로그 그룹이 고객 관리형 키로 암호화된 경우 키 정책은 CloudWatch Logs가 서비스 역할을 통해 KMS 키를 검증하고 CloudWatch Logs 서비스 보안 주체가 암호화 작업을 수행하도록 허용해야 합니다.

> **Important**
>
> AWS Security Agent는 사용자를 대신하여 CloudWatch Logs 로그 그룹 및 Secrets Manager 보안 암호를 생성할 수도 있습니다. KMS 키 정책을 구성할 때 이러한 서비스 생성 리소스에 해당하는 문도 포함합니다.

다음 키 정책 문은 CMK 암호화 리소스에 필요한 권한을 부여합니다. 구성에 적용되는 문만 포함합니다. 리소스가 에이전트 스페이스에 대해 지정한 것과 동일한 KMS 키를 사용하는 경우이 명령문을 해당 키의 정책에 추가합니다. 리소스가 다른 KMS 키를 사용하는 경우 대신 해당 키의 정책에 이러한 문을 추가합니다.

정책에서 다음 자리 표시자 값을 바꿉니다.

- *111122223333* - AWS 계정 ID
- *MyApplicationRole* - AWS Security Agent 설정 중에 생성된 애플리케이션 역할
- *MyPenTestServiceRole* - 침투 테스트 서비스 역할
- *us-east-1* - AWS Security Agent를 사용하는 AWS 리전

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKmsKeyAccessForCreatingSecrets",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyApplicationRole"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyDecryptionForS3Objects",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyPenTestServiceRole"
      },
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com",
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::YOUR-BUCKET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyDecryptionForSecretsManagerSecrets",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyPenTestServiceRole"
      },
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:YOUR-SECRET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyValidationForCloudWatchLogsLogGroups",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyPenTestServiceRole"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
```

```
        "StringLike": {
          "kms:ViaService": "logs.*.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyAccessForCloudWatchLogsLogGroups",
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.us-east-1.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333:log-group:YOUR-LOG-GROUP-NAME*"
        }
      }
    }
  ]
}
```

Note

- `AllowKmsKeyDecryptionForS3Objects` 문은 고객 관리형 키로 암호화된 S3 버킷에서 학습 리소스를 제공하는 경우에만 필요합니다. S3용 SSE-KMS 구성에 대한 자세한 내용은 SSE-KMS로 데이터 보호를 참조하세요.
- `AllowKmsKeyDecryptionForSecretsManagerSecrets` 문은 침투 테스트 자격 증명이 고객 관리형 키로 암호화된 Secrets Manager 보안 암호에 저장된 경우에만 필요합니다. 서비스 역할은 침투 테스트 중에 보안 암호를 해독할 수 있는 액세스 권한이 필요합니다. 보안 암호 암호화에 대한 자세한 내용은 Secrets Manager의 보안 암호 암호화 및 복호화를 참조하세요.

- 이 `AllowKmsKeyValidationForCloudWatchLogsLogGroups` 문은 서비스 역할을 부여`kms:DescribeKey`하므로 CloudWatch Logs는 서비스 역할을 통해 KMS 키를 로그 그룹과 연결할 때 KMS 키를 검증할 수 있습니다.
- 이 `AllowKmsKeyAccessForCreatingSecrets` 문은 웹 애플리케이션이 고객 관리형 키를 사용하여 사용자를 대신하여 Secrets Manager 보안 암호를 생성하는 경우에 필요합니다. 애플리케이션 역할에는 KMS 키로 보안 암호를 암호화하는 `kms:GenerateDataKey` `kms:Decrypt` 및가 필요합니다.
- `AllowKmsKeyAccessForCloudWatchLogsLogGroups` 문은 CloudWatch Logs 서비스 보안 주체(`logs.`*`us-east-1`*`.amazonaws.com`)에게 로그 데이터를 암호화하고 해독하는 데 필요한 권한을 부여합니다. 이 명령문은 AWS Security Agent가 기존 로그 그룹을 제공하지 않을 때 사용자를 대신하여 로그 그룹을 생성하는 경우에도 필요합니다. 자세한 내용은 AWS KMS를 사용하여 CloudWatch Logs에서 로그 데이터 암호화를 참조하세요.
- 여러 리소스가 동일한 KMS 키를 공유하는 경우 문을 단일 키 정책으로 결합할 수 있습니다.

관리자 역할

관리자 역할(콘솔에서 AWS Security Agent를 관리하는 데 사용하는 IAM 역할)에는 추가 IAM 정책이 필요하지 않습니다.

애플리케이션 역할

KMS 키 정책 외에도 AWS Security Agent 설정 중에 지정된 애플리케이션 역할에 다음 IAM 정책을 연결합니다. 이 정책은 역할에게 고객 관리형 키를 사용하여 에이전트 스페이스 데이터를 암호화 및 해독하고 AWS Secrets Manager에서 보안 암호를 생성할 수 있는 권한을 부여합니다.

정책에서 다음 자리 표시자 값을 바꿉니다.

- *`111122223333`* - AWS 계정 ID
- *`us-east-1`* - AWS Security Agent를 사용하는 AWS 리전
- 의 KMS 키 ARNs `Resource`- KMS 키의 ARNs

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSynchronousDataAccessForAgentSpaces",
      "Effect": "Allow",
```

```
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890cd"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333",
          "kms:ViaService": "securityagent.us-east-1.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:securityagent:agent-space":
 "arn:aws:securityagent:us-east-1:111122223333:agent-space/*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyAccessForCreatingSecrets",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890cd"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:*"
        }
      }
    }
  ]
}
```

> **Note**
>
> 애플리케이션 역할은 사용자가 지정하거나 AWS 보안 에이전트 설정 중에 콘솔이 생성하는 IAM 역할입니다. 웹 애플리케이션은이 역할을 수임하여 침투 테스트 실행 로그를 검색하고 사용자를 대신하여 Secrets Manager 보안 암호를 생성합니다.
>
> - `AllowKmsKeyAccessForCreatingSecrets` 문은 침투 테스트를 위한 인증 리소스를 구성하고 기존 보안 암호를 지정하는 대신 자격 증명을 직접 입력하도록 선택한 경우에 필요합니다. 웹 애플리케이션은 사용자를 대신하여 보안 암호를 생성하며, 에이전트 스페이스에 지정된 고객 관리형 키로 보안 암호를 암호화하려면 애플리케이션 역할에이 문의 권한이 필요합니다. 에이전트 스페이스에 사용되는 KMS 키와 일치하도록이 문의 `Resource` ARNs을 업데이트합니다.

서비스 역할

침투 테스트 중에 AWS Security Agent는 침투 테스트 서비스 역할을 수임하여 AWS 리소스에 액세스합니다. 이러한 리소스가 고객 관리형 키로 암호화된 경우 서비스 역할에 해당 KMS 키를 사용할 수 있는 권한을 부여해야 합니다. 이는 다음 리소스에 적용됩니다.

- S3 버킷 - 고객 관리형 키로 암호화된 S3 버킷에서 학습 리소스(예: API 문서, 위협 모델 또는 소스 코드)를 제공하는 경우 서비스 역할에는 해당 버킷의 객체를 복호화할 수 있는 권한이 필요합니다. S3용 SSE-KMS 구성에 대한 자세한 내용은 SSE-KMS로 데이터 보호를 참조하세요.
- Secrets Manager 보안 암호 - 침투 테스트 보안 인증 정보가 고객 관리형 키로 암호화된 Secrets Manager 보안 암호에 저장되는 경우 서비스 역할에는 해당 보안 암호를 해독할 수 있는 권한이 필요합니다. 보안 암호 암호화에 대한 자세한 내용은 Secrets Manager의 보안 암호 암호화 및 암호 해독을 참조하세요.
- CloudWatch Logs 로그 그룹 - 침투 테스트 실행 로그를 저장하는 데 사용되는 CloudWatch Logs 로그 그룹이 고객 관리형 키(사용자를 대신하여 AWS Security Agent에서 생성한 로그 그룹 포함)로 암호화된 경우 서비스 역할에 키를 검증할 `kms:DescribeKey` 권한이 필요합니다. CloudWatch Logs 서비스 보안 주체는 로그 데이터의 실제 암호화 및 복호화를 처리하며, 이러한 권한은 키 정책을 통해 부여됩니다(참조the section called “키 정책”). 로그 데이터 암호화에 대한 자세한 내용은 AWS KMS를 사용하여 CloudWatch Logs에서 로그 데이터 암호화를 참조하세요.

> **Important**
>
> 에이전트 스페이스에 지정한 것과 다른 KMS 키를 사용하여 이러한 리소스를 암호화하는 경우 서비스 역할이 침투 테스트 중에 액세스하는 리소스를 보호하는 각 KMS 키에 대해 서비스 역할 권한을 부여해야 합니다.

침투 테스트 서비스 역할에 다음 IAM 정책을 연결합니다. 구성에 적용되는 문만 포함합니다.

정책에서 다음 자리 표시자 값을 바꿉니다.

- *111122223333* - AWS 계정 ID
- *us-east-1* - AWS Security Agent를 사용하는 AWS 리전
- 의 KMS 키 ARNs `Resource`- 각 리소스를 암호화하는 데 사용되는 고객 관리형 키의 ARNs

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowKmsAccessForEncryptedS3Buckets",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE-S3-KEY-ID"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com",
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::YOUR-BUCKET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsAccessForEncryptedSecrets",
      "Effect": "Allow",
      "Action": [
```

```
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE-SECRETS-KEY-ID"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.*.amazonaws.com",
          "kms:EncryptionContext:SecretARN": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:YOUR-SECRET-NAME*"
        }
      }
    },
    {
      "Sid": "AllowKmsKeyValidationForCloudWatchLogsLogGroups",
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE-CLOUDWATCH-LOGS-KEY-ID"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "logs.*.amazonaws.com"
        }
      }
    }
  ]
}
```

Note

- `AllowKmsAccessForEncryptedS3Buckets` 문은 고객 관리형 키로 암호화된 S3 버킷에서 학습 리소스를 제공하는 경우에만 필요합니다. S3 버킷을 암호화하는 데 사용

되는 KMS 키와 일치하도록 `Resource` ARN을 업데이트하고 버킷 이름과 일치하도록 `kms:EncryptionContext:aws:s3:arn` 값을 업데이트합니다.

- `AllowKmsAccessForEncryptedSecrets` 문은 침투 테스트 자격 증명이 고객 관리형 키로 암호화된 Secrets Manager 보안 암호에 저장된 경우에만 필요합니다. 보안 암호를 암호화하는 데 사용되는 KMS 키와 일치하도록 `Resource` ARN을 업데이트하고 보안 암호 이름과 일치하도록 `kms:EncryptionContext:SecretARN` 값을 업데이트합니다.
- 명령`AllowKmsKeyValidationForCloudWatchLogsLogGroups`문은 CloudWatch Logs 로그 그룹이 사용자를 대신하여 AWS Security Agent에서 생성한 로그 그룹을 포함하여 고객 관리형 키로 암호화된 경우에만 필요합니다. 로그 그룹을 암호화하는 데 사용되는 KMS 키와 일치하도록 `Resource` ARN을 업데이트합니다.
- 여러 리소스가 동일한 KMS 키를 공유하는 경우 문을 결합하고 `Resource` 필드에 공유 키 ARN을 한 번 나열할 수 있습니다.

고객 관리형 키를 사용하여 리소스 생성

AWS Security Agent를 설정하거나 개별 리소스를 생성할 때 고객 관리형 키를 지정할 수 있습니다. KMS 키는 해당 리소스 및 해당 하위 리소스에 속하는 모든 데이터를 암호화합니다.

설정 중 기본 KMS 키 설정(콘솔)

초기 AWS Security Agent 설정 중에 기본 KMS 키를 구성할 수 있습니다. 이 키는 다른 키를 지정하지 않는 한 새 에이전트 스페이스 및 통합의 기본값으로 사용됩니다.

1. AWS Security Agent 설정 페이지에서 암호화 - 선택 사항 섹션을 확장합니다.
2. 암호화 설정 사용자 지정(고급)을 선택합니다.
3. AWS KMS 키 선택에 계정에서 KMS 키를 선택하거나 KMS 키 ARN을 입력합니다.
4. 나머지 설정 단계를 완료하고 AWS Security Agent 설정을 선택합니다.

AWS Security Agent는 KMS 키를 검증하여 KMS 키가 존재하는지, 활성화되었는지, 대칭 암호화를 사용하는지 확인합니다. 검증에 실패하면 설정이 진행되지 않고 오류 메시지가 표시됩니다.

고객 관리형 키를 사용하여 에이전트 공간 생성(콘솔)

1. AWS Security Agent 콘솔에서 에이전트 스페이스 페이지로 이동합니다.
2. 에이전트 공간 생성을 선택합니다.

3. 에이전트 스페이스의 이름과 선택적 설명을 입력합니다.
4. [Advanced] 섹션을 확장합니다.
5. 데이터 암호화에서 다음 중 하나를 선택합니다.
   - 애플리케이션 기본 키 사용 - AWS Security Agent 설정 중에 구성된 기본 KMS 키를 사용합니다. 기본 키가 구성된 경우이 옵션이 기본적으로 선택됩니다.
   - 다른 키 사용 -이 에이전트 스페이스에 대해 다른 KMS 키를 지정합니다. 암호화 설정 사용자 지정(고급)을 선택한 다음 AWS KMS 키 선택에 계정에서 KMS 키를 선택하거나 ARN을 입력합니다.
6. 생성(Create)을 선택합니다.

고객 관리형 키와의 통합 생성(콘솔)

1. AWS Security Agent 콘솔에서 통합 페이지로 이동합니다.
2. 통합 추가를 선택하고 공급자(예: GitHub)를 선택합니다.
3. 1단계 완료: 공급자 애플리케이션을 설치하고 승인합니다.
4. 2단계: 세부 정보 등록에서 등록 이름을 입력하고 공급자 설정을 구성합니다.
5. 데이터 암호화 섹션에서 암호화 설정 사용자 지정(고급)을 선택합니다.
6. AWS KMS 키 선택에 계정에서 KMS 키를 선택하거나 KMS 키 ARN을 입력합니다.
7. 연결을 선택합니다.

고객 관리형 키(AWS CLI 또는 SDK)를 사용하여 리소스 생성

AWS CLI 또는 SDK를 사용하여 고객 관리형 키를 지정하려면

- 애플리케이션의 기본 KMS 키를 설정`CreateApplication`하기 위해를 호출할 때 `defaultKmsKeyId` 파라미터를 포함합니다. 이 키는 명시적 KMS 키 없이 에이전트 스페이스 또는 통합을 생성할 때 대체 키로 사용됩니다.
- `CreateAgentSpace` 또는를 호출`CreateIntegration`할 때 `kmsKeyId` 파라미터를 포함하여 특정 리소스를 암호화합니다. 이는 애플리케이션 수준 기본값보다 우선합니다.

파라미터 세부 정보는 AWS Security Agent API 참조를 참조하세요.

KMS 키를 지정하지 않으면 AWS Security Agent는 애플리케이션 수준 기본 KMS 키가 구성된 경우 이를 사용합니다. 그렇지 않으면 리소스가 AWS관리형 키로 암호화됩니다.

키 교체

AWS Security Agent는 정기적으로 브랜치 키를 자동으로 교체합니다. 브랜치 키 교체는 모든 이전 버전을 유지하면서 브랜치 키의 새 활성 버전을 생성합니다. 교체 후:

- 새 데이터는 새 브랜치 키 버전으로 암호화됩니다.
- 이전에 암호화된 데이터는 이전 브랜치 키 버전을 사용하여 복호화할 수 있습니다.
- 데이터 재암호화는 필요하지 않습니다.

브랜치 키 교체는 KMS 키 교체와는 별개입니다. 고객 관리형 키에 대해 자동 KMS 키 교체를 독립적으로 활성화할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 KMS 키 교체를 참조하세요.

브랜치 키 교체 후에도 새 암호화 작업에 이전 브랜치 키의 캐시된 복사본을 계속 사용할 수 있는 짧은 기간이 있습니다. 이 기간은 내부 캐시 TTL(time-to-live)에 의해 결정되며 데이터 보안에 영향을 주지 않습니다.

고려 사항

AWS Security Agent에서 고객 관리형 키를 사용할 때는 다음 사항에 유의하세요.

- 고객 관리형 키는 새 리소스에만 적용됩니다. 고객 관리형 키를 구성하기 전에 생성된 기존 리소스는 계속해서 AWS관리형 암호화를 사용합니다. 기존 데이터의 마이그레이션은 없습니다.
- 고객 관리형 키는 생성 시 지정됩니다. 리소스를 생성할 때 KMS 키를 지정합니다. 기존 리소스에 대한 KMS 키는 추가하거나 변경할 수 없습니다.
- 여러 KMS 키가 지원됩니다. 리소스마다 다른 KMS 키를 사용할 수 있습니다. 예를 들어 프로덕션 에이전트 스페이스에는 하나의 키를 사용하고 개발 에이전트 스페이스에는 다른 키를 사용할 수 있습니다.
- KMS 키를 비활성화하거나 삭제하면 데이터에 액세스할 수 없게 됩니다. KMS 키의 삭제를 비활성화하거나 예약하면 AWS Security Agent는 해당 키로 암호화된 데이터를 해독할 수 없습니다. 영향을 받는 리소스는 키가 다시 활성화될 때까지 액세스할 수 없게 됩니다. KMS 키를 삭제하면 해당 키로 암호화된 모든 데이터에 영구적으로 액세스할 수 없습니다.
- 키 정책 권한이 필요합니다. KMS 키 정책에서 필요한 권한을 제거하면 AWS Security Agent는 암호화된 데이터에 액세스할 수 없습니다. 키 정책이에 설명된 대로 관리자 역할(AWS 콘솔에서 서비스를 관리하는 데 사용됨), 애플리케이션 역할(웹 애플리케이션에서 사용됨), 침투 테스트 서비스 역할(에이전트가 침투 테스트를 실행하기 위해 수임함) 및 AWS Security Agent 서비스 보안 주체에 대한 권한을 부여하는지 확인합니다the section called “필수 KMS 권한”.

- AWS KMS 할당량이 적용됩니다. KMS 키에 대한 암호화 작업은 AWS KMS 요청 할당량에 포함됩니다. 일반적인 사용에서 계층적 키링 아키텍처는 브랜치 키 캐싱을 통해 KMS 호출을 최소화합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 할당량 요청을 참조하세요.

문제 해결

AWS Security Agent를 사용할 때 일반적으로 나타나는 오류에 대한 솔루션을 찾습니다.

액세스 거부: 잘못된 GitHub 계정 유형이 선택되었거나 잘못된 조직 이름이 지정됨

- 원하는 GitHub 조직에 AWS Security Agent 애플리케이션을 설치했지만 `User` 대신를 `GitHub Account Type` 로 잘못 설정했습니다. `Organization`
- AWS Security Agent 애플리케이션을 원하는 GitHub 조직에 설치하고를 `GitHub Account Type` 로 올바르게 설정`Organization`했지만 `Organization Name` 필드를 비워 두었거나 애플리케이션을 설치한 조직과 일치하지 않는 잘못된 조직 이름을 입력했습니다.

해결 방법:

1. GitHub로 이동하여 조직에서 앱을 제거합니다. 자세한 내용은 the section called “1단계: GitHub에서 AWS 보안 에이전트 GitHub 앱 제거” 단원을 참조하십시오.
2. 통합 페이지로 돌아가서를 클릭하여 통합 프로세스를 다시 시작하고 앱을 원하는 GitHub 조직에 다시 `Add Integrations`설치하고 권한을 부여합니다.
3. 의 드롭다운`Organization`에서를 선택합니다. `GitHub account type`
4. `Organization Name` 입력한가 애플리케이션을 설치한 EXACT와 동일한지 확인합니다.
5. 연결을 선택하여 GitHub 조직 통합을 생성합니다.

액세스 거부: 조직에 GitHub 앱을 설치할 수 있는 권한 부족

원하는 GitHub 조직에 AWS Security Agent 애플리케이션을 설치하려고 하면 설치 페이지의 버튼에 두 가지 메시지가 표시됩니다.

- 조직은 `Member` 다음을 볼 수 있습니다. `Authorize & Request`
- 조직은 `Owner` 다음을 볼 수 있습니다. `Install & Authorize`

아래 단계에 따라 GitHub 조직의 인지 Member 또는 Owner 인지 확인할 수 있습니다.

1. github.com 이동합니다.
2. 웹 사이트에서 프로필 선택
3. Organizations 드롭다운 메뉴에서 로 이동하여 선택합니다.
4. 조직 목록에서 AWS Security Agent를 설치할 조직을 찾습니다. 그러면 조직 이름 Owner 옆에가 있는지 아니면 Member가 있는지 지정됩니다.

가능한 해결책:

- 통합을 생성하기 전에 소유자가 설치 요청을 승인하도록 합니다.
- 소유자가 GitHub 조직의 역할을에서 로 업데이트Owner하고 통합 프로세스를 다시 시작Member하도록 합니다.

침투 테스트 중에 에이전트가 엔드포인트에 연결할 수 없음

침투 테스트 에이전트가 구성된 대상 URL을 호출할 수 없거나 대상 엔드포인트를 성공적으로 탐색하지 못하는 경우:

- 엔드포인트가 구성된 대상 URL 외부의 도메인을 호출하는 경우 추가 도메인이 Pentest 구성에 액세스 가능한 URLs로 추가되었는지 확인합니다.
- 침투 테스트는 현재 포트 80 또는 443에서 트래픽을 처리하는 HTTP/HTTPS 엔드포인트에만 사용할 수 있습니다.
- WAF가 구성된 경우 WAF가 침투 테스트 트래픽을 차단하지 않는지 확인합니다. securityagent 헤더별로 침투 테스트 트래픽을 허용할 수 있으며User-Agent, 기본적으로 로 설정됩니다.

추가 도움말 받기

다음 문제 해결 단계를 시도한 후에도 문제가 계속 발생하는 경우:

- AWS Security Agent 서비스 상태 페이지 확인
- 복잡한 구성 문제에 대한 지원은 AWS Support에 문의하세요.

사용자 및 개발자용(웹 앱 및 IDE)

웹 애플리케이션, IDE 또는 연결된 소스 공급자에서 설계 검토, 위협 모델, 코드 검토 및 침투 테스트를 실행한 다음 결과를 검토하고 해결합니다.

AWS Security Agent 웹 앱에 로그인

AWS Security Agent 웹 앱에 로그인하여 다음과 같은 작업을 완료해야 합니다.

- 설계 검토 수행
- 침투 테스트 수행

액세스 방법

AWS Security Agent는 조직이 초기 설정 중에 액세스를 구성한 방식과 AWS 콘솔 액세스 권한 여부에 따라 웹 애플리케이션에 액세스하는 다양한 방법을 제공합니다.

IAM Identity Center(SSO) - 조직에서 IAM Identity Center를 활성화한 경우 SSO 자격 증명을 사용하여 로그인할 수 있습니다. 웹 애플리케이션에 액세스하려면 먼저 IAM Identity Center에서 하나 이상의 에이전트 스페이스에 할당되어야 합니다.

관리자 액세스(IAM) - 적절한 권한이 있는 AWS 콘솔 액세스 권한이 있는 경우 에이전트 스페이스 페이지의 관리자 액세스 링크를 통해 웹 애플리케이션을 시작할 수 있습니다. 이 방법은 기존 콘솔 세션을 통해 인증을 제공합니다.

> **Note**
>
> 조직에서 사용하는 기본 액세스 방법은 초기 AWS Security Agent 설정 중에 결정되었습니다. 사용자 액세스 및 할당 구성에 대한 자세한 내용은 섹션을 참조하세요the section called “사용자에게 AWS Security Agent 웹 앱에 대한 액세스 권한 부여”.

IAM Identity Center(SSO)로 로그인

조직에서 IAM Identity Center 액세스를 구성한 경우 여러 진입점을 통해 SSO 자격 증명을 사용하여 웹 애플리케이션에 로그인할 수 있습니다.

사전 조건

- IAM Identity Center에서 하나 이상의 에이전트 스페이스에 할당되었습니다.
- IAM Identity Center SSO 자격 증명이 있는 경우

웹 애플리케이션에 액세스

필요에 따라 다음 방법 중 하나를 선택합니다.

할당된 모든 에이전트 스페이스를 보려면:

1. AWS Security Agent 콘솔에서 설정으로 이동합니다.
2. 웹 애플리케이션 도메인을 찾아 URL을 복사합니다.

> **Tip**
>
> 쉽게 액세스할 수 있도록이 URL을 북마크합니다. 할당된 모든 에이전트 스페이스를 볼 수 있는 범용 진입점입니다.

3. 웹 애플리케이션 URL로 이동합니다.
4. 메시지가 표시되면 IAM Identity Center 자격 증명을 입력합니다.
5. 인증 후 할당된 모든 에이전트 스페이스의 목록이 표시됩니다.
6. 작업할 에이전트 스페이스를 선택합니다.

특정 에이전트 스페이스에 직접 액세스하려면(AWS 콘솔 액세스 필요):

1. AWS Management Console에 로그인합니다.
2. AWS Security Agent 콘솔로 이동합니다.
3. 액세스하려는 에이전트 공간으로 이동합니다.
4. 다음 중 하나를 선택합니다.
   - 에이전트 스페이스 개요 페이지의 웹 애플리케이션 시작 버튼
   - 코드 검토 섹션의 웹 앱 시작 버튼
   - 침투 테스트 섹션의 웹 앱 시작 버튼
5. 메시지가 표시되면 IAM Identity Center 자격 증명을 입력합니다.
6. 인증 후 웹 애플리케이션의 해당 에이전트 스페이스로 바로 이동합니다.

> **Note**
>
> AWS 콘솔 액세스 권한이 없는 경우 설정 페이지의 웹 애플리케이션 도메인을 사용하여 할당된 모든 에이전트 스페이스에 액세스합니다.

관리자 액세스(IAM)로 로그인

적절한 AWS Security Agent 권한이 있는 AWS 콘솔 액세스 권한이 있는 경우 자동 인증이 포함된 관리자 액세스 링크를 통해 웹 애플리케이션을 시작할 수 있습니다.

사전 조건

- AWS Management Console에 로그인했습니다.
- AWS Security Agent 리소스에 액세스할 수 있는 적절한 권한이 있음

웹 애플리케이션에 액세스

다음 방법 중 한 가지를 선택하세요.

특정 에이전트 스페이스에 액세스하려면:

1. AWS Management Console에 로그인합니다.
2. AWS Security Agent 콘솔로 이동합니다.
3. 액세스하려는 에이전트 공간으로 이동합니다.
4. 에이전트 스페이스 개요 페이지에서 관리자 액세스 버튼을 선택합니다.
5. 웹 애플리케이션은 해당 에이전트 스페이스에 대한 자동 인증과 함께 새 탭에서 열립니다.

> **Note**
>
> 조직에서 IAM Identity Center를 기본 액세스 방법으로 사용하는지 여부에 관계없이 AWS 콘솔 액세스 권한이 있는 사용자는 항상 관리자 액세스 버튼을 사용할 수 있습니다.

설계 검토 생성

AWS Security Agent가 검토할 파일을 업로드하여 조직 보안 요구 사항을 기준으로 설계 문서를 평가합니다. 설계 검토를 통해 개발 수명 주기 초기에 보안 문제를 식별할 수 있으므로 가장 비용 효율적인 아키텍처 문제를 해결할 수 있습니다.

AWS Security Agent는 조직의 보안 요구 사항을 기준으로 설계 문서를 분석하여 구현이 시작되기 전에 보안 태세를 개선하기 위한 자세한 보안 조사 결과를 제공합니다.

이 절차에서는 분석을 위해 설계 파일을 업로드하여 설계 검토를 생성합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 업로드할 준비가 된 문서 설계(DOC, DOCX, JPEG, MD, PDF, PNG 및 TXT)
- 각 파일은 2MB 이하여야 하며, 모든 파일에서 총 6MB를 합산해야 합니다.
- 조직에 대해 활성화된 보안 요구 사항 이해

1단계: 설계 검토 생성 시작

에이전트 웹 앱에서 설계 검토 생성 페이지로 이동합니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 설계 검토 섹션으로 이동합니다.
3. 디자인 검토 생성을 선택합니다.

> **Tip**
>
> AWS Security Agent 콘솔의 보안 요구 사항 페이지로 이동하여 조직의 활성화된 보안 요구 사항을 볼 수 있습니다. 세부 정보를 보려면 활성화된 요구 사항을 선택합니다. 이러한 요구 사항은 설계 파일을 분석하는 데 사용됩니다.

2단계: 설계 검토 이름 지정

이 설계 검토의 목적과 범위를 식별하는 데 도움이 되는 설명 이름을 제공합니다.

1. 디자인 검토 이름 섹션에서 이름 필드를 찾습니다.
2. 설계 검토에 대한 설명이 포함된 이름을 입력합니다.

> **Note**
>
> 이름은 검토 중인 프로젝트, 기능 또는 구성 요소를 명확하게 식별해야 합니다. 최대 80자입니다.

3단계: 설계 파일 업로드

AWS Security Agent가 보안 규정 준수를 위해 분석할 설계 문서를 업로드합니다.

1. 검토할 파일 섹션에서 파일 요구 사항을 검토합니다.

> **Important**
>
> 설계 검토당 최대 5개의 파일을 업로드할 수 있습니다. 각 파일은 2MB 이하여야 하며, 모든 파일에서 총 6MB를 합산해야 합니다. 지원되는 형식: DOC, DOCX, JPEG, MD, PDF, PNG 및 TXT.

2. 다음 방법 중 하나를 사용하여 파일을 업로드합니다.
   a. 끌어서 놓기 - 파일을 파일 드롭 영역 영역으로 직접 드래그합니다.
   b. 찾아보기 - 파일 선택 버튼을 사용하여 컴퓨터에서 파일을 찾아보고 선택합니다.
3. 필요한 모든 파일이 업로드되었는지 확인합니다.

> **Tip**
>
> 최상의 결과를 얻으려면 시스템의 보안 관련 구성 요소 및 데이터 흐름을 설명하는 아키텍처 다이어그램, 설계 사양 및 기술 설명서를 포함하세요.

4단계: 설계 검토 시작

모든 필수 정보를 구성한 후 설계 문서의 보안 분석을 시작합니다.

1. 업로드된 모든 파일과 설정을 검토하여 정확성을 확인합니다.
2. 설계 검토 시작을 선택합니다.
3. AWS Security Agent는 활성화된 보안 요구 사항을 기준으로 설계 문서를 분석합니다.

> **Note**
>
> 설계 검토 프로세스는 일반적으로 업로드된 파일의 수와 크기에 따라 몇 분 내에 완료됩니다. 조직의 보안 요구 사항에 따라 보안 조사 결과를 받게 됩니다.

다음 단계

설계 검토를 시작한 후:

- 에이전트 웹 앱에서 검토 진행 상황 모니터링
- 보안 결과 검토
- 개발 팀과 조사 결과 공유
- 설계에서 식별된 보안 조사 결과 해결
- 설계 문서를 업데이트하고 필요한 경우 다시 제출

설계 검토의 결과 검토

조사 결과를 검토하면 어떤 보안 요구 사항이 충족되는지, 어떤 주의를 기울여야 하는지, 구현이 시작되기 전에 설계의 보안 태세를 개선하기 위해 어떤 조치를 취해야 하는지 이해하는 데 도움이 됩니다.

이 절차에서는 설계 검토 조사 결과를 액세스, 필터링 및 해석하여 보안 조사 결과를 효과적으로 해결하는 방법을 알아봅니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 완료된 설계 검토

- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 조직의 활성화된 보안 요구 사항에 대한 지식

1단계: 설계 검토에 액세스

설계 검토로 이동하여 결과 및 요약 정보를 봅니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 설계 검토 섹션으로 이동합니다.
3. 목록에서 검사할 설계 검토를 선택합니다.

> **Tip**
>
> 설계 검토 세부 정보 페이지에는 검토 상태, 완료 날짜 및 검토된 파일 수에 대한 요약이 표시됩니다.

2단계: 결과 요약 검토

상위 수준 요약을 검토하여 설계의 전반적인 보안 태세를 이해합니다.

1. 요약 섹션을 찾습니다.
2. 규정 준수, 규정 미준수, 데이터 부족, 해당 사항 없음 등 각 규정 준수 상태 범주의 수를 검토합니다.

> **Note**
>
> 요약은 각 상태 유형에 대한 수를 제공하므로 주의가 필요한 조사 결과 수를 빠르게 평가할 수 있습니다. 각 상태에 대한 자세한 설명은 4단계를 참조하세요.

3단계: 조사 결과 필터링 및 탐색

필터링 및 검색 기능을 사용하여 특정 조사 결과 또는 규정 준수 상태에 집중할 수 있습니다.

1. 결과 검토 섹션에서 필터 컨트롤을 찾습니다.
2. 상태별로 필터링하려면:

a. 상태 드롭다운 메뉴를 선택합니다.

b. 특정 규정 준수 상태를 선택하여 해당 상태의 조사 결과만 봅니다.

3. 특정 보안 요구 사항을 검색하려면:

a. 검색 필드에 키워드를 입력합니다.

b. 입력하면 결과가 자동으로 업데이트됩니다.

4. 페이지 매김 컨트롤을 사용하여 조사 결과의 여러 페이지를 탐색합니다.

> **Tip**
>
> 먼저 규정 미준수 상태를 기준으로 필터링하여 설계에 즉각적인 주의가 필요한 조사 결과의 우선순위를 지정합니다.

4단계: 규정 준수 상태 이해

각 결과에는 설계가 특정 보안 요구 사항을 해결하는 방법을 나타내는 규정 준수 상태가 표시됩니다.

- 규정 준수 - 설계가 분석을 기반으로 보안 요구 사항을 충족합니다.
- 규정 미준수 - 설계가 보안 요구 사항을 위반하거나 부적절하게 해결합니다.
- 데이터 부족 - 업로드된 파일에는 규정 준수를 확인할 수 있는 충분한 정보가 없습니다.
- 해당 사항 없음 - 보안 요구 사항은 시스템 설계에 적용되지 않습니다.

> **Important**
>
> 규정 미준수 및 데이터 부족 상태는 조치가 필요하므로 이에 중점을 둡니다. 설계를 업데이트하여 규정 미준수 조사 결과를 해결하고 추가 설계 설명서를 업로드하여 부족한 데이터 조사 결과를 해결합니다.

5단계: 결과 세부 정보 보기

개별 조사 결과를 선택하여 자세한 정당화 및 수정 지침을 확인합니다.

1. 조사 결과 테이블에서 보안 요구 사항 이름을 선택합니다.
2. 다음을 포함한 결과 세부 정보를 검토합니다.

- 평가 중인 특정 보안 요구 사항
- 누락되거나 규정을 준수하지 않는 항목에 대한 구체적인 세부 정보를 포함하여 조사 결과가 규정 준수 상태를 받은 이유를 설명하는 설명
- 조사 결과를 해결하기 위한 권장 해결 지침
- 보안 요구 사항에 대한 조직의 내부 설명서 또는 표준에 대한 링크

> **Note**
>
> 이 주석은 AWS Security Agent의 추론과 구체적인 세부 정보를 설명합니다. 데이터 조사 결과가 충분하지 않은 경우 주석은 "설계 문서에서 인증 메커니즘을 언급하지 않음" 또는 "저장 데이터 암호화에 대한 정보를 찾을 수 없음"과 같이 누락된 정보를 식별합니다.

6단계: 조사 결과 해결

설계의 보안 태세를 개선하기 위해 주의가 필요한 결과에 대해 조치를 취합니다.

규정 미준수 조사 결과의 경우:

1. 권장 문제 해결 지침을 검토합니다.
2. 연결된 내부 설명서에서 추가 컨텍스트를 검토합니다.
3. 보안 요구 사항을 해결하도록 설계 문서를 업데이트합니다.
4. 나중에 참조할 수 있도록 변경 사항을 문서화합니다.

데이터 조사 결과 부족의 경우:

1. 설명을 주의 깊게 읽고 어떤 특정 정보가 누락되었는지 이해하세요.
2. 누락된 세부 정보로 설계 문서를 생성하거나 업데이트합니다.
3. 업데이트된 파일을 다시 제출할 준비를 합니다.

다음 단계

설계 조사 결과를 검토한 후:

- 조사 결과 보고서를 팀과 공유할 CSV 파일로 다운로드합니다.
- 규정 미준수 조사 결과를 처리하도록 설계 문서 업데이트

- 데이터 조사 결과 부족에 대한 추가 설명서 생성
- 논의를 위해 개발 팀과 조사 결과 공유
- 이 설계 검토를 복제하여 미리 로드된 원본 문서로 새 검토를 생성하면 이름을 업데이트하고 분석을 다시 실행하여 개선 사항을 확인할 수 있습니다.
- 규정 준수 요구 사항을 충족하는 설계에 대한 구현 진행

보안 요구 사항 관리에 대한 자세한 내용은 섹션을 참조하세요the section called “보안 요구 사항 관리”.

설계 검토 생성에 대한 자세한 내용은 섹션을 참조하세요the section called “설계 검토 생성”.

위협 모델 생성

AWS Security Agent 웹 애플리케이션에서 위협 모델을 생성하여 애플리케이션의 아키텍처를 분석하고 보안 위협을 식별합니다. 위협 모델은 두 가지 주요 출력을 생성합니다. 하나는 시스템이 구축되고 작동하는 방식을 설명하는 시스템 개요이고, 다른 하나는 심각도 수준, STRIDE 분류 및 실행 가능한 권장 사항이 있는 시스템을 공격할 수 있는 방법을 설명하는 위협 세트입니다.

위협 모델은 두 가지 종류의 입력을 허용하며, 다음 중 하나 또는 둘 다를 제공할 수 있습니다.

- 범위 문서 - 위협 모델의 초점을 정의하는 기술 설계 문서, API 사양 또는 아키텍처 문서입니다. 제공된 에이전트는 이러한 문서에 설명된 컨텍스트로 분석 범위를 지정합니다. 통합을 통해 파일(DOC, DOCX, JPEG, MD, PDF, PNG, TXT)을 업로드하거나, S3 URIs를 제공하거나, Confluence 페이지를 연결할 수 있습니다.
- 소스 - 연결된 리포지토리(GitHub, GitHub Enterprise Server, GitLab, GitLab 자체 관리형 또는 Bitbucket) 또는 S3 위치의 소스 코드입니다. AWS Security Agent는 기존 시스템을 이해하기 위해 소스 코드를 읽습니다. 범위 문서와 결합하면 소스 코드는 현재 구현에 대한 컨텍스트를 제공합니다.

이 절차에서는 입력 및 권한을 구성하여 위협 모델을 생성한 다음 실행을 시작합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 다음 중 최소 하나 이상을 지정해야 합니다.
  - 소스로 사용할 연결된 리포지토리(GitHub, GitHub Enterprise Server, GitLab, GitLab 자체 관리형 또는 Bitbucket)(참조the section called “위협 모델링 활성화”)

- 소스 코드가 포함된 S3 버킷
- 범위 문서 또는 Confluence 통합으로 업로드할 문서 설계

> **Tip**
>
> 에이전트 스페이스에 이미 연결된 리포지토리 또는 S3 버킷이 있는 경우(예: 코드 검토 또는 침투 테스트 설정을 통해) 위협 모델을 즉시 생성할 수 있으므로 추가 설정이 필요하지 않습니다.

AWS Security Agent가 위협 모델을 실행하는 방법

제공하는 입력에 따라 AWS Security Agent가 위협 모델을 실행하는 방법이 결정됩니다. 세 가지 시나리오가 있습니다.

입력한 입력	AWS Security Agent가 위협 모델을 실행하는 방법
소스 전용(소스 코드)	AWS Security Agent는 소스 코드를 분석하여 애플리케이션의 구조를 이해하고, 구성 요소를 분류하고, 데이터 흐름을 추출하고, 위협 모델을 구축하고, 시스템이 실제로 구현된 방식에 따라 위협을 식별합니다.
범위 문서만(설계 문서)	AWS Security Agent는 문서에 설명된 설계에 초점을 맞춘 위협 모델을 실행합니다. 범위 문서에 설명된 아키텍처, 데이터 흐름 및 구성 요소를 기반으로 위협을 식별합니다. 코드가 존재하기 전에 설계를 위협 모델링하는 데 유용합니다.
소스 및 범위 문서 모두	AWS Security Agent는 범위 문서에 설명된 컨텍스트(예: 새 기능에 대한 설계 문서)로 위협 모델의 범위를 지정합니다. 소스 코드를 사용하여 기존 시스템을 파악한 다음 위협은 해당 설계가 이미 구현되었는지 여부에 관계없이 범위 문서에 설명된 설계를 모델링합니다.

위협 모델 페이지에 액세스

웹 애플리케이션의 위협 모델링 섹션으로 이동합니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 왼쪽 사이드바에서 위협 모델을 선택합니다.

3. 제목, 최신 실행 상태 및 심각도별 위협 수와 함께 기존 위협 모델 목록이 표시됩니다.

위협 모델 생성

입력 및 권한을 구성하여 새 위협 모델을 설정합니다.

1. 위협 모델 페이지에서 위협 모델 생성을 선택합니다.

위협 모델 세부 정보 구성

위협 모델의 제목을 제공합니다.

1. 제목 필드에 위협 모델을 설명하는 이름을 입력합니다.

> **Tip**
>
> 모델링할 애플리케이션 또는 서비스를 식별하는 이름을 사용합니다. 예: "billing-service" 또는 "checkout-api".

범위 문서 추가

아키텍처 설명서, API 사양 또는 설계 제안과 같이 위협 모델의 초점을 정의하는 기술 설계 문서를 제공합니다. 범위 문서가 제공되면 에이전트는 이러한 문서에 설명된 컨텍스트로 분석 범위를 지정합니다. 소스를 제공하는 경우 범위 문서는 선택 사항입니다.

1. 범위 문서 섹션에서 다음 방법 중 하나 이상을 사용하여 문서를 추가합니다.
   - 파일 업로드 - 설계 문서를 직접 업로드합니다(DOC, DOCX, JPEG, MD, PDF, PNG 또는 TXT).
   - S3 URIs- 연결된 S3 버킷에 저장된 문서를 가리키는 S3 URIs를 입력합니다.
   - Confluence 페이지 - 에이전트 스페이스에 Confluence 통합이 연결되어 있는 경우 Confluence 워크스페이스에서 페이지를 선택합니다.

> **Tip**
>
> 최상의 결과를 얻으려면 시스템의 구성 요소, 데이터 흐름 및 신뢰 경계를 설명하는 아키텍처 다이어그램, API 사양 및 기술 설명서를 포함하세요.

소스 추가

AWS Security Agent가 기존 시스템을 이해하는 데 사용하는 소스 코드를 선택합니다. 범위 문서와 결합하면 소스 코드는 현재 구현에 대한 컨텍스트를 제공합니다. 에이전트는 이를 사용하여 이미 존재하는 것을 이해합니다. 범위 문서를 제공하는 경우 소스는 선택 사항입니다.

1. 소스 섹션에서 다음 방법 중 하나 이상을 사용하여 소스 코드를 추가합니다.

통합 리포지토리

에이전트 스페이스(GitHub, GitHub Enterprise Server, GitLab, GitLab 자체 관리형 또는 Bitbucket)에 연결된 리포지토리 중에서 선택합니다.

1. 소스로 포함하려는 각 리포지토리 옆의 확인란을 선택합니다.
2. 검색 필드를 사용하여 이름으로 특정 리포지토리를 찾습니다.

> **Note**
>
> 에이전트 스페이스에 연결된 리포지토리만 여기에 표시됩니다. 리포지토리를 더 추가하려면 관리자에게 에이전트 공간 구성을 업데이트하도록 요청합니다.

S3 소스

연결된 S3 버킷에 저장된 소스 코드를 가리키는 S3 URIs를 추가합니다.

1. 포함할 각 소스 코드 위치에 대한 S3 URI를 입력합니다.

AWS 리소스 구성

이 위협 모델에 대한 IAM 서비스 역할과 선택적 CloudWatch 로그 그룹을 선택합니다.

1. 서비스 역할 드롭다운의 구성된 서비스 역할에서 IAM 역할을 선택합니다.

> **Note**
>
> 서비스 역할에는 S3의 소스 코드에 액세스하고 CloudWatch 로그에 쓸 수 있는 권한이 있어야 합니다. 서비스 역할은 AWS 관리 콘솔에서 에이전트 스페이스를 설정하는 동안 구성됩니다.

2. (선택 사항) 로그 그룹 드롭다운에서 위협 모델 실행 로그를 저장할 CloudWatch 로그 그룹을 선택합니다.

위협 모델 생성

1. 구성을 검토합니다.
2. 위협 모델 생성을 선택합니다.

실행을 시작할 수 있는 위협 모델 세부 정보 페이지로 리디렉션됩니다.

위협 모델 실행

위협 모델을 생성한 후 실행을 시작하여 분석을 시작합니다.

1. 위협 모델 세부 정보 페이지에서 실행 시작을 선택합니다.
2. AWS Security Agent는 소스 및 범위 문서를 분석하기 시작합니다.

실행하려는 위협 모델 옆에 있는 실행 시작을 선택하여 위협 모델 목록 페이지에서 실행을 시작할 수도 있습니다.

위협 모델 실행 모니터링

위협 모델이 실행될 때 진행 상황을 추적합니다.

실행 상태

실행 페이지 헤더에는 상태 표시기가 표시됩니다.

- 진행 중 - 위협 모델 에이전트가 실행 중입니다.
- 중지 중 - 취소가 요청되었습니다. 에이전트가 중지하기 전에 현재 작업을 완료하고 있습니다.
- 완료됨 - 실행이 성공적으로 완료되었습니다.

- 실패 - 실행에 오류가 발생했습니다. 오류 메시지가 세부 정보와 함께 표시됩니다. 문제를 해결한 후 새 실행을 시작합니다.
- 중지됨 - 사용자가 실행을 취소했습니다. 중지 전에 생성된 모든 위협은 보존됩니다.

실행 페이지 탭

실행 세부 정보 페이지에서 탭 간에 이동합니다.

- 개요 - 심각도별 위협 분석(중요, 높음, 중간, 낮음)을 보여주는 심각도 수준 파이 차트와 함께 실행 요약(작업 ID, 시작 시간, 상태, 기간)을 봅니다. 요약 아래에서 각 STRIDE 범주에 속하는 위협 수를 보여주는 STRIDE 위협 범주 막대 차트를 봅니다. 실행이 완료되면 에이전트가 생성한 시스템 개요를 봅니다.
- 구성 -이 실행에 사용된 소스, 문서 및 권한(서비스 역할, CloudWatch 로그 그룹)을 봅니다.
- 사전 처리 - 인프라 설정 로깅, S3 소스 액세스 검증(해당하는 경우), 환경 설정 테스트 등 위협 분석이 시작되기 전에 실행되는 사전 처리 검사의 상태를 확인합니다. 진행률 표시줄에는 완료된 검사 수가 표시됩니다.
- 로그 - 실행 중에 에이전트가 수행한 작업의 필터링 가능한 목록을 봅니다. 사이드 패널에서 세부 로그 출력을 보려면 작업을 선택합니다.
- 위협 - 실행 중에 식별된 위협을 봅니다(참조the section called “위협 모델의 위협 검토”).

실행 기록

각 위협 모델은 모든 실행 기록을 유지합니다. 위협 모델 세부 정보 페이지에서:

- 실행 테이블에는 시작 시간, 상태, 기간, 위협 수 및 ID가 포함된 모든 실행이 나열됩니다.
- 실행의 시작 시간 링크를 선택하여 전체 세부 정보를 봅니다.

> **Tip**
>
> 소스 코드를 업데이트하거나 범위 문서를 수정한 후 동일한 위협 모델을 다시 실행하여 시간 경과에 따른 시스템 개요 및 위협의 변화를 확인합니다.

위협 모델 편집

위협 모델 구성을 수정하려면:

1. 위협 모델 세부 정보 페이지에서 편집을 선택합니다.
2. 필요에 따라 제목, 소스, 범위 문서 또는 AWS 리소스를 업데이트합니다.
3. 변경 사항 저장을 선택합니다.

> **Note**
>
> 위협 모델을 편집해도 이전에 완료된 실행에는 영향을 주지 않습니다. 실행 시 사용된 구성 스냅샷을 보존합니다.

위협 모델 삭제

위협 모델과 관련된 모든 실행 및 위협을 삭제하려면:

1. 위협 모델 세부 정보 페이지에서 작업 메뉴를 열고 삭제를 선택합니다.
2. 삭제를 확인합니다.

> **Important**
>
> 실행이 현재 진행 중인 경우 위협 모델을 삭제하기 전에 실행을 중지해야 합니다.

다음 단계

위협 모델을 실행한 후:

- 시스템 개요 및 위협 검토(참조the section called “위협 모델의 위협 검토”)
- 위협이 해결되었는지 확인하기 위해 변경한 후 위협 모델을 다시 실행합니다.
- 애플리케이션이 발전함에 따라 소스 및 범위 문서 조정

위협 모델의 위협 검토

위협 모델 실행이 완료되면 시스템 개요와 위협을 검토하여 애플리케이션이 공격될 수 있는 방식과 이에 대해 수행할 작업을 파악합니다. 시스템 개요는 애플리케이션의 아키텍처, 신뢰 경계, 데이터 흐름 및 보안 태세를 설명하는 포괄적인 문서입니다. 각 위협에는 문, 심각도 수준, STRIDE 분류, 영향을 받는 자산 및 이에 대한 해결 권장 사항이 포함됩니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 완료된 위협 모델 실행
- AWS Security Agent 웹 애플리케이션에 대한 액세스

1단계: 위협 모델 실행에 액세스

완료된 위협 모델 실행으로 이동합니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 왼쪽 사이드바에서 위협 모델을 선택합니다.
3. 검사하려는 위협 모델을 선택합니다.
4. 실행 테이블에서 시작 시간 링크를 선택하여 완료된 실행을 선택합니다.

2단계: 시스템 개요 검토

시스템 개요는 AWS Security Agent가 시스템을 이해하는 방법에 대한 포괄적인 설명입니다. 다음을 포함할 수 있는 구조화된 문서입니다.

- 목적 - 시스템이 수행하는 작업 및 서비스 대상입니다.
- 기능 - 시스템이 제공하는 주요 기능입니다.
- 설계 의도 - 위협 모델링 중인 설계 변경 또는 기능입니다(범위 문서가 제공되는 경우).
- 아키텍처 - 배포 패턴 및 통신 프로토콜을 포함하여 시스템을 구축하는 방법입니다.
- 구성 요소 - 용도와 주요 상호 작용이 포함된 시스템 구성 요소의 테이블입니다.
- 신뢰 경계 - 각 교차 시 존재하는 보호를 포함하여 보안 컨텍스트가 변경되는 위치입니다.
- 데이터 흐름 - 각 단계의 프로토콜, 자격 증명 및 보호를 포함하여 데이터가 시스템을 통해 이동하는 방식에 대한 자세한 설명입니다.
- 보안 태세 - 현재 인증, 암호화 및 액세스 제어 메커니즘입니다.
- 민감한 자산 - 분류 및 노출 지점과 함께 보호가 필요한 데이터 및 자격 증명입니다.
- 주요 가정 - 에이전트가 시스템에 대해 한 보안 관련 가정입니다.

시스템 개요를 검토하려면:

1. 개요 탭을 선택합니다.
2. 작업 ID, 시작 시간, 상태 및 기간을 보여주는 실행 요약 섹션을 검토합니다.
3. 심각도 수준 차트를 검토합니다.이 차트는 심각도(심각, 높음, 중간, 낮음)별 위협의 분류를 개수 및 백분율과 함께 보여줍니다.
4. 각 STRIDE 범주(스푸핑, 변조, 거부, 정보 공개, 서비스 거부, 권한 상승)에 속하는 위협 수를 보여주는 위협 범주 차트를 검토합니다.
5. 시스템 개요 섹션에서 에이전트의 전체 분석을 읽습니다.

> **Tip**
>
> 시스템 개요가 시스템을 정확하게 반영하지 않는 경우 입력을 구체화하고, 관련 리포지토리를 소스로 추가하거나 더 완전한 범위 문서를 업로드하고, 위협 모델을 다시 실행합니다.

3단계: 위협 검토

위협 탭으로 이동하여 실행 중에 식별된 모든 위협을 확인합니다.

1. 위협 탭을 선택합니다.
2. 위협은 각 카드에 위협 제목, 심각도 배지, 상태 및 마지막으로 업데이트된 타임스탬프가 표시된 목록으로 표시됩니다. 심각도, 상태 또는 제목별 검색을 기준으로 위협을 필터링할 수 있습니다.
3. 목록에서 위협을 선택하여 오른쪽 패널에서 전체 세부 정보를 봅니다.

위협 심각도

각 위협에는 심각도 수준이 할당됩니다.

- 심각 - 즉각적인 조치가 필요합니다. 악용 시 전체 시스템 손상이 발생할 수 있습니다.
- 높음 - 즉각적인 주의가 필요하며, 악용 시 상당한 보안 영향이 발생할 수 있습니다.
- 중간 - 적절한 기간 내에 해결해야 하며 전반적인 보안 위험에 기여합니다.
- 낮음 - 정기 유지 관리의 일부로 해결할 수 있으며 즉각적인 위험을 최소화합니다.

위협 세부 정보

위협을 선택하여 오른쪽 패널에서 세부 정보를 봅니다. 세부 정보는 다음 섹션으로 구성되어 있습니다.

메타데이터 행:

- 심각도 - 에이전트가 할당한 위험 수준(심각, 높음, 중간 또는 낮음).
- STRIDE 범주 - 위협 분류: 스푸핑, 변조, 거부, 정보 공개, 서비스 거부 또는 권한 상승.
- 생성 시점 - 위협이 식별된 시점입니다.
- 최종 업데이트 - 위협이 마지막으로 수정된 시간입니다.

설명 섹션:

- 문 - 위협에 대한 자연어 설명: 위협 소스가 수행할 수 있는 작업, 영향, 이를 가능하게 하는 조건.
- 소스 - 위협의 액터 또는 오리진(예: “인증된 사용자” 또는 “외부 공격자”)입니다.
- 작업 - 위협 소스가 수행할 수 있는 작업(예: "검색 파라미터에 SQL 쿼리 삽입").
- 영향 - 위협 조치의 직접적인 결과(예: "고객 레코드에 대한 무단 액세스").

참조 섹션:

- 영향을 받는 자산 - 위협의 영향을 받는 특정 자산(예: "고객 결제 레코드" 또는 "DynamoDB 테이블").
- 사전 조건 - 위협이 악용될 수 있으려면 true여야 하는 조건입니다.
- 영향을 받는 목표 - 영향을 받는 보안 목표: 기밀성, 무결성, 가용성, 권한 부여, 인증 또는 거부 금지.
- 증거 - 위협을 지원하는 소스 코드 파일 경로로, 리포지토리의 특정 파일에 다시 연결됩니다.
- 앵커 - 위협을 소스 코드의 특정 노드에 연결하는 코드 참조입니다.

권장 사항 섹션:

- 권장 사항 - 위협 해결을 위한 실행 가능한 지침입니다.

4단계: 수동으로 위협 생성

수동 검토 중에 또는 외부 소스에서 발견된 위협과 같이 에이전트가 식별하지 못한 위협을 추가할 수 있습니다.

1. 완료된 실행의 위협 탭에서 위협 생성을 선택합니다.
2. 위협 세부 정보를 입력합니다.
   - 문 - 위협에 대한 자연어 설명입니다.

- 심각도 - 심각, 높음, 중간 또는 낮음을 선택합니다.
- 소스 - 위협의 액터 또는 오리진입니다.
- 사전 조건 - 위협을 악용하는 데 필요한 조건입니다.
- 작업 - 위협 소스가 수행할 수 있는 작업입니다.
- 영향 - 위협 조치의 직접적인 결과입니다.
- 영향을 받는 자산 - 영향을 받는 특정 자산(쉼표로 구분).
- 영향을 받는 보안 목표 - 기밀성, 무결성, 가용성, 권한 부여, 인증 또는 비거부 중에서 선택합니다.
- STRIDE 범주 - 적용 가능한 범주를 선택합니다.
- 권장 사항 - 위협 해결을 위한 지침입니다.

3. 생성(Create)을 선택합니다.

수동으로 생성된 위협은 에이전트가 생성한 위협과 함께 위협 목록에 나타납니다.

5단계: 위협 편집 및 분류

위협을 검토할 때 세부 정보를 편집하고 상태를 업데이트하여 진행 상황을 추적할 수 있습니다.

1. 목록에서 위협을 선택합니다.
2. 위협 세부 정보 패널에서 편집 아이콘을 선택하여 편집 양식을 엽니다.
3. 다음 필드를 수정할 수 있습니다.
   - 상태 - 위협 수명 주기 추적:
     - 열기 - 위협이 승인되고 주의가 필요합니다(기본값).
     - 해결됨 - 문제를 해결했습니다.
     - 무시됨 - 위협을 검토하고 해당 사항이 없다고 판단했습니다.
   - 심각도 - 에이전트의 평가가 컨텍스트와 일치하지 않는 경우 심각도 수준을 조정합니다.
   - 문 - 위협 설명을 구체화합니다.
   - 소스, 사전 조건, 작업, 영향 - 도메인 지식을 기반으로 위협 세부 정보를 업데이트합니다.
   - 영향을 받는 자산 - 영향을 받는 자산을 추가하거나 제거합니다(쉼표로 구분).
   - 영향을 받는 보안 목표 - 영향을 받는 보안 목표(기밀성, 무결성, 가용성, 권한 부여, 인증, 비거부)를 선택합니다.
4. 저장을 선택하여 변경 사항을 적용합니다.

6단계: 보고서 다운로드

실행이 완료되면 시스템 개요와 식별된 모든 위협을 요약한 PDF 보고서를 다운로드할 수 있습니다.

1. 완료된 실행 페이지에서 보고서 생성을 선택합니다.
2. PDF가 컴퓨터로 다운로드됩니다.

7단계: 사전 검사 및 로그 검토

에이전트가 어떻게 결론에 도달했는지 조사하거나 부분적인 실패를 디버깅해야 하는 경우:

1. 사전 처리 탭을 선택하여 위협 분석이 시작되기 전에 실행되는 사전 처리 검사의 상태를 확인합니다. 각 점검에는 상태(완료, 진행 중 또는 보류 중)가 표시되고 진행률 표시줄에는 완료된 점검 수가 표시됩니다. 완료된 검사를 확장하여 세부 로그 출력을 볼 수 있습니다.
2. 로그 탭을 선택하여 실행 중에 에이전트가 수행한 작업의 필터링 가능한 목록을 봅니다. 목록에서 작업을 선택하여 측면 패널에서 세부 로그 출력을 봅니다.

다음 단계

위협 모델 결과를 검토한 후:

- 에이전트의 권장 사항에 따라 심각도가 높은 위협을 먼저 해결합니다.
- 수정 사항을 구현할 때 위협 상태 업데이트
- 새 위협 모델을 실행하여 변경 사항이 식별된 위협을 해결하는지 확인합니다.
- 애플리케이션이 발전함에 따라 소스 및 범위 문서를 조정합니다(참조the section called “위협 모델 생성”).

풀 요청에서 코드 보안 조사 결과 검토

리포지토리의 풀 요청에 대한 코드 검토를 활성화한 후 AWS Security Agent는 풀 요청을 자동으로 분석하고 소스 제어 공급자에게 직접 보안 조사 결과를 게시합니다. 이를 통해 개발자는 풀 요청을 벗어나지 않고도 일반 워크플로 내에서 보안 문제를 해결할 수 있습니다.

> **Note**
>
> 이 페이지는 GitHub 풀 요청, GitLab 병합 요청 및 Bitbucket 풀 요청에 적용됩니다. 경험은 모든 공급자에서 비슷합니다.

풀 요청에서 코드 검토 작동 방식

코드 검토가 활성화된 리포지토리에서 풀 요청(또는 GitLab에서 병합 요청)을 제출하면 AWS Security Agent가 자동으로 분석을 시작합니다.

1. 풀 요청 분석 트리거 - 코드 검토 기능을 활성화한 리포지토리에서 풀 요청이 "검토 준비 완료"로 표시되면 코드 검토가 트리거됩니다. 초안 풀 요청은 분석되지 않습니다.
2. 분석 승인 - AWS Security Agent가 풀 요청을 분석하기 시작하면 "AWS Security Agent가 코드를 분석 중입니다... " 이렇게 하면 분석이 시작되었고 진행 중임을 알 수 있습니다.
3. 검토 완료 - 분석이 완료되면 AWS Security Agent는 결과와 함께 풀 요청에 검토를 게시합니다. 모든 보안 조사 결과는 단일 검토로 일괄 처리되어 풀 요청을 정리하고 알림을 최소화합니다.

코드 검토 결과 이해

AWS Security Agent는 분석 중에 찾은 내용에 따라 다양한 유형의 결과를 제공합니다.

보안 문제가 발견된 경우

AWS Security Agent는 코드 변경에서 보안 문제를 식별하면 다음을 포함하는 검토를 게시합니다.

- 요약 - 식별된 문제의 유형과 잠재적 영향을 설명하는 모든 보안 조사 결과에 대한 개략적인 개요
- 개별 조사 결과 - 세부 보안 조사 결과는 기본 검토 아래에 스레드 주석으로 표시되며 각 조사 결과는 다음을 포함합니다.
  - 보안 문제에 대한 설명
  - 문제가 발견된 코드의 위치
  - 문제 해결 방법을 설명하는 해결 지침
  - 코드 검토 설정에 따른 관련 컨텍스트(보안 요구 사항 위반, 일반적인 취약성 또는 둘 다)

> **Note**
>
> 분석되는 보안 문제의 유형은 코드 검토 설정에 따라 다릅니다. 보안 요구 사항 검증을 구성한 경우 조사 결과는 조직의 사용자 지정 보안 요구 사항을 참조합니다. 보안 취약성 조사 결과를 구성한 경우 조사 결과는 일반적인 보안 취약성을 식별합니다. 코드 검토 설정에 대한 자세한 내용은 섹션을 참조하세요the section called “GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화”.

보안 문제가 발견되지 않은 경우

AWS Security Agent가 분석을 완료하고 코드 변경에서 보안 문제를 찾지 못하면 "No issues identified"라는 주석이 게시됩니다. 이렇게 하면 검토가 성공적으로 완료되고 코드 변경으로 구성된 코드 검토 설정에 따라 보안 조사 결과가 트리거되지 않았습니다.

보안 조사 결과에 대한 대응

AWS Security Agent에서 게시한 보안 조사 결과를 검토한 후 소스 제어 공급자에서 직접 조치를 취할 수 있습니다.

- 조사 결과 해결 - 조사 결과에 제공된 수정 지침에 따라 코드를 업데이트한 다음 새 커밋을 풀 요청에 푸시합니다. AWS Security Agent는 업데이트된 코드를 분석합니다.
- 대화 해결 - 보안 조사 결과를 해결한 후 대화를 해결됨으로 표시하여 진행 상황을 추적합니다.

> **Tip**
>
> 각 결과에는 식별된 보안 문제에 맞는 특정 수정 지침이 포함되어 있습니다. 이 지침을 주의 깊게 검토하여 보안 위험과 이를 효과적으로 해결하는 방법을 이해합니다.

코드 검토 조사 결과 필터링

리포지토리에 `filtering.md` 파일을 추가하여 AWS Security Agent가 코드를 분석하는 방법을 사용자 지정할 수 있습니다. 이 파일을 사용하면 코드베이스에 대한 컨텍스트를 제공하고 분석에서 파일 또는 폴더를 제외하여 거짓 긍정을 줄일 수 있습니다.

필터링 파일 생성

리포지토리 루트의 .awssecurityagent 디렉터리filtering.md에 라는 파일을 생성합니다.

```
.awssecurityagent/filtering.md
```

AWS Security Agent는 풀 요청을 분석할 때 리포지토리의 기본 브랜치(예: main 또는 mainline)에서 이 파일을 읽습니다.

파일 구조

filtering.md 파일은 AWS Security Agent가 인식하는 특정 섹션과 함께 표준 마크다운 형식을 사용합니다. 파일에는 Code Review 제목 뒤에 IgnorePatterns 및 ContextHints 섹션 중 하나 또는 둘 다(공백 없음)가 포함되어야 합니다.

다음 예제에서는 filtering.md 파일의 전체 구조를 보여줍니다.

```
# filtering.md

## Code Review

### IgnorePatterns

**/*.md

/myapp/src/**/*.snap

/myapp/config/README

### ContextHints

- The backend is a trusted system and won't return non-standard protocols.
- URL is generated from server with presigned token, so no SSRF security
 vulnerabilities.
- AppSec has verified that we are allowed to use cache with an eviction policy.
```

패턴 무시

이 IgnorePatterns 섹션에서는 코드 검토 중에 AWS Security Agent가 건너뛰어야 하는 파일과 폴더를 지정합니다. glob patterns를 사용하여 분석에서 제외할 경로를 정의합니다.

형식 지정 요구 사항

- 각 패턴은 자체 줄에 있어야 합니다.
- 각 패턴을 그 사이에 빈 줄로 구분합니다. 이렇게 하면 GitHub 또는 코드 검토 도구에서 볼 때 파일이 올바르게 렌더링됩니다.
- 패턴은 표준 glob 형식을 따릅니다. 예를 들어 `**/*.md`는 모든 마크다운 파일과 일치하고 루트의 `/myapp/src/` 폴더 내 모든 `.snap` 파일과 `/myapp/src/**/*.snap` 일치합니다.
- 이 섹션에서는 최대 1,000개의 무시 패턴을 지원합니다.

컨텍스트 힌트

이 `ContextHints` 섹션에서는 AWS Security Agent가 보다 정확한 평가를 수행하는 데 도움이 되는 코드베이스에 대한 추가 컨텍스트를 제공합니다. 컨텍스트 힌트를 사용하여 아키텍처 결정, 보안 예외 또는 결과 해석 방식에 영향을 미칠 수 있는 기타 정보를 설명합니다.

형식 지정 요구 사항

- 각 힌트는 대시(-)와 공백으로 시작해야 합니다.
- 각 힌트를 500자로 제한된 자유 형식 텍스트의 한 줄로 작성합니다.
- 각 힌트는 코드베이스에 대한 특정 컨텍스트를 설명해야 합니다.
- 이 섹션에서는 최대 20개의 컨텍스트 힌트를 지원합니다.

컨텍스트 힌트는 AWS Security Agent가 초기 분석을 완료한 후 적용되므로 특정 사용 사례에 적용되지 않는 결과를 필터링하는 데 도움이 됩니다.

다음 단계

코드 보안 조사 결과를 검토한 후:

- 문제 해결 지침에 따라 코드 업데이트
- 새 커밋을 푸시하여 변경 사항에 대한 재분석 트리거
- 필요한 경우 코드 검토 설정 조정(참조the section called “GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화”)
- 조직의 보안 요구 사항을 검토하여 검증 기준 이해
- 배포된 애플리케이션의 포괄적인 보안 검증을 위한 침투 테스트 고려

코드 검토 생성

AWS Security Agent 웹 애플리케이션에서 코드 검토를 생성하여 소스 코드 리포지토리 및 S3 소스에서 보안 취약성을 스캔합니다. 코드 검토는 전체 코드베이스에서 포괄적인 정적 분석을 수행하여 보안 문제를 식별하고 문제 해결 지침을 제공합니다.

개별 코드 변경 사항을 분석하는 풀 요청 기반 코드 검토(참조the section called “풀 요청에서 코드 보안 조사 결과 검토”)와 달리 온디맨드 코드 검토는 전체 소스 코드를 스캔하여 보안 취약성을 식별하고 조직의 보안 요구 사항 준수를 검증합니다.

이 절차에서는 소스 코드 입력을 선택하고, 권한을 구성하고, 검토를 실행하여 코드 검토를 생성합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 에이전트 스페이스에 연결된 GitHub 리포지토리 또는 S3 버킷 하나 이상

> **Tip**
>
> 이미 에이전트 스페이스에 연결된 GitHub 리포지토리가 있는 경우 코드 검토를 사용할 준비가 되어 있으므로 추가 설정이 필요하지 않습니다. 에이전트 스페이스 페이지의 코드 검토 카드에서 웹 앱에서 시작을 선택하거나 웹 애플리케이션을 직접 시작합니다.
> 추가 소스를 연결하거나 S3 버킷을 구성해야 하는 경우 섹션을 참조하세요the section called “코드 검토 활성화”.

코드 검토 페이지에 액세스

웹 애플리케이션의 코드 검토 섹션으로 이동합니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 왼쪽 사이드바에서 코드 검토를 선택합니다.
3. 소스 정보, 마지막 실행 상태 및 결과 요약과 함께 기존 코드 검토 목록이 표시됩니다.

코드 검토 생성

소스 코드 입력 및 권한을 구성하여 새 코드 검토를 설정합니다.

1. 코드 검토 페이지에서 코드 검토 생성을 선택합니다.

코드 검토 세부 정보 구성

제목을 입력하고 검토할 소스 코드를 선택합니다.

1. 제목 필드에 코드 검토를 설명하는 이름을 입력합니다.

> **Tip**
>
> 애플리케이션, 리포지토리 또는 검토 범위를 식별하는 이름을 사용합니다. 예: "billing-service-security-review" 또는 "infrastructure-code-audit".

2. 소스 섹션에서이 검토를 위한 소스 코드 입력을 선택합니다. 다음 두 가지 탭에서 선택합니다.

GitHub 리포지토리

에이전트 스페이스에 연결된 리포지토리 중에서 선택합니다.

1. GitHub 리포지토리 탭을 선택합니다.
2. 통합 리포지토리 테이블에서 검토에 포함할 각 리포지토리 옆의 확인란을 선택합니다.
3. 검색 필드를 사용하여 이름으로 특정 리포지토리를 찾습니다.

> **Note**
>
> 코드 검토 구성을 통해 에이전트 스페이스에 연결된 리포지토리만 여기에 표시됩니다. 리포지토리를 더 추가하려면 관리자 콘솔에서 관리를 선택하거나 관리자에게 에이전트 공간 구성을 업데이트하도록 요청합니다.

S3 소스

에이전트 스페이스에 연결된 S3 버킷에서 ZIP 파일을 선택합니다. 에이전트 스페이스 관리자는 사용 가능한 S3 버킷을 구성합니다. 이러한 버킷 중 하나에 저장된 모든 ZIP 파일을 코드 검토의 소스로 사용할 수 있습니다.

1. S3 소스 탭을 선택합니다.
2. 검토에 포함할 각 ZIP 파일의 S3 URI를 입력합니다. 최대 30개의 S3 소스를 추가할 수 있습니다.

> **Note**
>
> S3 소스는 에이전트 스페이스에 연결된 S3 버킷에 저장된 ZIP 파일이어야 합니다. 추가 버킷을 사용하려면 섹션을 참조하세요the section called “코드 검토 활성화”.

권한 구성

이 코드 검토를 위해 IAM 서비스 역할과 선택적 CloudWatch 로그 그룹을 선택합니다.

1. 권한 섹션에서 서비스 역할 드롭다운을 찾습니다.
2. 구성된 서비스 역할에서 IAM 역할을 선택합니다.

> **Note**
>
> 서비스 역할에는 S3의 소스 코드에 액세스하고 CloudWatch 로그 및 코드 검토에 필요한 기타 AWS 리소스에 쓸 수 있는 권한이 있어야 합니다. 서비스 역할은 AWS Management Console에서 코드 검토 설정 중에 구성됩니다.

3. (선택 사항) CloudWatch 로그 그룹 드롭다운에서 코드 검토 실행 로그를 저장할 로그 그룹을 선택합니다.

> **Note**
>
> 로그 그룹을 선택하지 않으면 AWS Security Agent는 코드 검토 로그를 저장하기 위한 기본 로그 그룹을 생성합니다.

자동 코드 수정 구성

AWS Security Agent가 검토가 완료되는 즉시 모든 결과에 대한 코드 수정을 생성하도록 자동 수정을 활성화합니다.

1. 자동 코드 수정 섹션에서 자동 코드 수정 활성화 확인란을 선택합니다.

AWS Security Agent가 수정 사항을 제공하는 방법은 소스에 따라 다릅니다.

- 프라이빗 GitHub 리포지토리 - AWS Security Agent는 수정 사항이 포함된 풀 요청을 리포지토리에 제출합니다.
- 퍼블릭 GitHub 리포지토리 - 취약성이 수정되기 전에 공개되지 않도록 AWS Security Agent는 풀 요청을 열지 않습니다. 대신 웹 애플리케이션에서 다운로드하여 비공개로 적용할 수 있는 결과에 제안된 차이를 연결합니다.
- S3 소스 - 코드 수정을 사용할 수 없습니다. 조사 결과 세부 정보를 검토하고 수정 사항을 수동으로 적용합니다.

> **Important**
>
> 프라이빗 리포지토리에 제출된 수정 풀 요청은 읽기 액세스 권한이 있는 모든 사용자에게 표시됩니다. 병합하기 전에 변경 사항을 검토합니다. 자동 코드 수정은 GitHub 리포지토리를 소스로 선택한 경우에만 사용할 수 있습니다.

> **Note**
>
> 비활성화된 경우에도 검토가 완료된 후 개별 GitHub 소스 결과에 대한 코드 수정을 수동으로 트리거할 수 있습니다.

시뮬레이션된 검증 구성

시뮬레이션된 검증을 활성화하여 실행 중인 애플리케이션에서 검색된 취약성이 악용 가능한지 동적으로 확인합니다.

1. 시뮬레이션된 검증 섹션에서 시뮬레이션된 검증 활성화 확인란을 선택합니다.

활성화되면 AWS Security Agent는 정적 분석이 완료된 후 시뮬레이션된 환경을 프로비저닝합니다. 소스 코드를 온보딩하고, 환경 내에서 서비스를 시작하고, 스캔 중에 발견된 취약성을 악용하려고 시도합니다. 그런 다음 취약성이 성공적으로 악용되었는지 여부를 나타내는 검증 상태로 결과가 업데이트됩니다.

> **Note**
>
> 시뮬레이션된 검증은 현재 독립형 도킹 가능 애플리케이션에서만 사용할 수 있습니다. 여러 리포지토리를 소스로 선택하면 시뮬레이션된 검증을 사용할 수 없습니다.

> **Important**
>
> 시뮬레이션된 검증은 코드 검토 실행에 처리 시간을 추가합니다. 검증 단계는 환경을 프로비저닝하고 악용 시도를 실행합니다. 조사 결과 수와 애플리케이션 복잡성에 따라 일반적으로 1~3시간이 걸립니다.

허용된 네트워크 대상

시뮬레이션된 환경에는 사전 정의된 허용 목록으로 제한된 아웃바운드 네트워크 액세스가 있습니다. 애플리케이션은 검증 중에 다음 도메인에 도달할 수 있습니다.

다음 표에는 허용되는 도메인과 그 용도가 나와 있습니다.

도메인	용도
`.amazonaws.com` , `.aws.amazon.com`	AWS 서비스
`.public.ecr.aws`	Amazon ECR 퍼블릭
`.docker.com` , `.docker.io`	Docker Hub
`.github.com` , `.githubusercontent.com`	GitHub
`.gitlab.com`	GitLab

도메인	용도
.npmjs.com , .npmjs.org	npm 레지스트리
.pypi.org , .pypi.python.org , .pythonhosted.org	Python Package Index
.crates.io , .rustup.rs	Rust 패키지
.maven.org , .gradle.org	Java/Gradle 패키지
.nuget.org	.NET 패키지
.rubygems.org , .ruby-lang.org	Ruby 패키지
.golang.org , .pkg.go.dev , .goproxy.io	Go 패키지
.nodejs.org , .yarnpkg.com	Node.js
.alpinelinux.org , .debian.org , .ubuntu.com , .centos.org , .fedoraproject.org	Linux 배포 리포지토리
.cloudfront.net	CloudFront 배포
.google.com , .googleapis.com	Google APIs
.microsoft.com , .visualstudio.com	Microsoft 서비스

도메인	용도
.sourceforge.net , .bitbucket.org	소스 호스팅

> **Note**
>
> 애플리케이션에이 목록에 없는 도메인에 대한 네트워크 액세스가 필요한 경우 네트워크 방화벽에 의해 연결이 차단됩니다. 외부 APIs 또는 위에 나열되지 않은 서비스에 의존하는 애플리케이션은 시뮬레이션된 환경에서 올바르게 시작되지 않을 수 있습니다.

코드 검토 생성

1. 구성을 검토하여 정확성을 확인합니다.
2. 코드 검토 생성을 선택합니다.

검토 실행을 시작할 수 있는 코드 검토 세부 정보 페이지로 리디렉션됩니다.

코드 검토 실행

코드 검토를 생성한 후 실행을 시작하여 분석을 시작합니다.

1. 코드 검토 세부 정보 페이지에서 검토 시작을 선택합니다.
2. AWS Security Agent가 소스 코드 분석을 시작합니다.

실행하려는 코드 검토 옆에 있는 검토 시작을 선택하여 코드 검토 목록 페이지에서 검토를 시작할 수도 있습니다.

코드 검토 실행 모니터링

코드 검토가 실행될 때 코드 검토 진행 상황을 추적합니다.

실행 단계 검토

코드 검토 실행은 진행률 지표로 표시되는 다음 단계를 거칩니다.

1. Preflight - AWS Security Agent는 소스 코드에 대한 액세스를 검증하고 테스트 환경을 설정합니다. 비행 전 검사에는 다음이 포함됩니다.
   - 서비스 인프라 설정
   - S3 소스 액세스 검증
   - 설정 테스트 환경
2. 정적 분석 - AWS Security Agent는 소스 코드에서 보안 취약성 및 요구 사항 위반을 검사합니다.
3. 시뮬레이션된 검증(선택 사항) - 시뮬레이션된 검증이 활성화된 경우 AWS Security Agent는 시뮬레이션된 환경을 프로비저닝하고 애플리케이션을 배포합니다. 그런 다음 정적 분석 중에 발견된 취약성을 악용하려고 시도합니다. 이 단계에서는 조사 결과가 대상 런타임에서 악용 가능한지 여부를 확인합니다. 이 단계는 코드 검토 생성 중에 시뮬레이션된 검증을 활성화한 경우에만 나타납니다.
4. 완료 - AWS Security Agent가 결과를 컴파일하고 결과 요약을 생성합니다.

실행 세부 정보 보기

실행 세부 정보 페이지에서 탭 사이를 탐색하여 진행 상황을 모니터링합니다.

- 코드 검토 실행 - 실행 ID, 생성 시간, 상태, 기간, 작업 시간, 심각도 수준 분석 및 위험 유형 차트를 포함한 실행 요약을 봅니다.
- 사전 처리 - 각 검증 단계의 사전 확인 진행 상황 및 상태를 확인합니다.
- 코드 검토 로그 - AWS Security Agent가 검토 중에 식별하고 수행한 작업을 각 단계에 대한 자세한 작업 로그와 함께 봅니다.
- 시뮬레이션된 검증 - 시뮬레이션된 검증이 활성화되면 프로비저닝 상태, 개별 결과에 대한 검증 작업 및 결과를 확인합니다.
- 조사 결과 - 검토가 완료된 후 보안 조사 결과를 봅니다(참조the section called “코드 검토의 결과 검토”).

실행 기록

각 코드 검토는 모든 실행 기록을 유지합니다. 코드 검토 세부 정보 페이지에서:

- 최신 실행 섹션에는 시작 시간, 상태, 기간 및 ID와 함께 가장 최근 실행이 표시됩니다.
- 모든 실행 테이블에는 시작 시간, 상태, 기간, 결과 요약 및 ID와 함께 이전 실행이 모두 나열됩니다.
- 최신 활성 실행의 세부 정보를 보려면 실행 모니터링을 선택합니다.
- 실행의 시작 시간 링크를 선택하여 전체 세부 정보를 봅니다.

다음 단계

코드 검토를 실행한 후:

- 보안 조사 결과 및 문제 해결 지침 검토(참조the section called “코드 검토의 결과 검토”)
- 자동 풀 요청 또는 수동 수정을 통해 조사 결과 해결(참조the section called “코드 검토 결과 수정”)
- 수정 사항을 구현한 후 추가 검토를 실행하여 문제 해결 확인
- 코드베이스가 발전함에 따라 코드 검토 구성 또는 소스 조정

코드 검토의 결과 검토

코드 검토 실행이 완료되면 실행 요약 및 보안 조사 결과를 검토하여 소스 코드의 취약성을 파악합니다. 각 결과에는 보안 문제의 우선 순위를 지정하고 해결하는 데 도움이 되는 설명, 심각도 등급, 코드 위치, 위험 추론 및 제안된 수정 사항이 포함되어 있습니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 완료되었거나 진행 중인 코드 검토 실행
- AWS Security Agent 웹 애플리케이션에 대한 액세스

1단계: 코드 검토 실행에 액세스

완료된 코드 검토 실행으로 이동하여 요약 및 조사 결과를 봅니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 왼쪽 사이드바에서 코드 검토를 선택합니다.
3. 검사하려는 코드 검토를 선택합니다.
4. 모든 실행 테이블에서 시작 시간 링크를 클릭하여 완료된 실행을 선택합니다. 또는 실행 모니터링을 선택하여 최신 실행을 봅니다.

2단계: 실행 진행 상황 모니터링

단계 표시기를 사용하여 코드 검토 실행 진행 상황을 추적합니다.

1. 페이지 헤더 아래에서 가로 단계 표시기를 찾습니다.

2. 각 단계의 상태를 검토합니다.
   - Preflight - 소스 코드에 대한 액세스를 검증하고 스캔 환경을 설정합니다. 검사에는 서비스 인프라 설정, S3 소스 액세스 검증 및 GitHub 액세스 검사를 포함하는 테스트 환경 설정이 포함됩니다.
   - 정적 분석 - 소스 코드에서 보안 취약성 및 요구 사항 위반을 스캔합니다.
   - 시뮬레이션된 검증(선택 사항) - 활성화하면 시뮬레이션된 환경을 프로비저닝하고 실행 중인 애플리케이션에 대해 발견된 취약성을 악용하려고 시도합니다.
   - 완료 - 결과를 컴파일하고 결과 요약을 생성합니다.

> **Note**
>
> 각 단계에는 상태 표시기(완료됨 또는 진행 중)가 표시됩니다. 모든 단계에 완료됨이 표시되면 실행이 완료된 것입니다. 시뮬레이션된 검증 단계는 코드 검토 생성 중에 시뮬레이션된 검증을 활성화한 경우에만 나타납니다.

3단계: 실행 요약 검토

코드 검토 실행 탭으로 이동하여 상위 수준 결과를 확인합니다.

1. 코드 검토 실행 탭을 선택합니다.
2. 실행 요약 섹션에서는 실행 상태, 기간 및 기타 상위 수준 세부 정보를 제공합니다. 또한 심각도 수준 및 위험 유형별로 분류된 보안 조사 결과의 대시보드를 제공합니다.
3. AWS Security Agent의 애플리케이션 개요는 실행이 완료될 때 코드 검토 스캔의 요약을 제공합니다.

4단계: 사전 처리 결과 검토

Preflight 탭으로 이동하여 모든 소스 액세스 검사가 통과되었는지 확인합니다.

1. Preflight 탭을 선택합니다.
2. 완료된 점검 수를 보여주는 Preflight 진행률 표시기를 검토합니다.
3. 각 점검에 성공 상태가 표시되는지 확인합니다.
   - 서비스 인프라 설정 - 테스트 환경이 준비되었는지 확인합니다.
   - S3 소스 액세스 검증 - S3 소스 코드에 대한 액세스를 확인합니다(해당하는 경우).
   - 테스트 환경 설정 - 분석 환경이 구성되어 있는지 확인합니다.

> **Note**
>
> 사전 검사에 실패하면 실행이 정적 분석으로 진행되지 않습니다. 검사 세부 정보를 검토하여 액세스 문제를 식별하고 해결한 다음 새 실행을 시작합니다.

5단계: 코드 검토 로그 검토

코드 검토 로그 탭으로 이동하여 분석 중에 AWS Security Agent가 수행한 작업을 확인합니다.

1. 코드 검토 로그 탭을 선택합니다.
2. 왼쪽 패널에서 작업 목록을 찾습니다.
3. 작업을 선택하여 오른쪽 패널에서 세부 로그를 봅니다.

> **Tip**
>
> 검색 필드를 사용하여 이름을 기준으로 작업을 필터링합니다. 로그는 AWS Security Agent가 조사한 내용과 결론에 도달한 방식에 대한 통찰력을 제공합니다.

6단계: 조사 결과로 이동

결과 탭을 선택하여 실행의 모든 보안 결과를 봅니다.

> **Note**
>
> 기본 신뢰도 필터
>
> 기본적으로 에이전트 신뢰도가 높은 결과만 표시됩니다. 중간 또는 낮은 에이전트 신뢰도와 거짓 긍정으로 조사 결과를 표시하려면 확인되지 않은 조사 결과 숨기기 토글을 끕니다.

1. 조사 결과 탭을 선택합니다.
2. 조사 결과는 왼쪽에 조사 결과 목록이 있고 오른쪽에 선택한 조사 결과의 세부 정보가 있는 분할 보기에 표시됩니다.
3. 각 결과 카드에 표시된 정보를 검토합니다.

- 조사 결과 제목 - 보안 문제를 요약하는 설명이 포함된 이름입니다.
- 심각도 배지 - 색상으로 구분된 심각도 지표:
  - 심각(빨간색) - 즉각적인 조치가 필요합니다. 악용 시 시스템 손상으로 이어질 수 있습니다.
  - 높음(빨간색) - 즉각적인 주의가 필요합니다. 악용 시 상당한 보안 영향이 발생할 수 있습니다.
  - 중간(주황색) - 적절한 기간 내에 해결해야 합니다. 전반적인 보안 위험에 기여합니다.
  - 낮음(노란색) - 정기 유지 관리의 일부로 해결할 수 있으며 즉각적인 위험을 최소화합니다.
- 최종 업데이트 - 조사 결과가 마지막으로 업데이트된 시점의 타임스탬프

7단계: 결과 세부 정보 검토

개별 조사 결과를 선택하여 각 취약성에 대한 포괄적인 정보를 봅니다.

1. 왼쪽 패널에서 결과를 선택하여 오른쪽 패널에 세부 정보를 표시합니다.
2. 사용 가능한 작업을 검토합니다.
   - 결과 해결 - 해결한 후 결과를 해결된 것으로 표시합니다.
   - 코드 수정 - 수정을 사용하여 풀 요청 생성(GitHub 소스에 사용 가능)
3. 이 결과가 정확했나요?를 사용하여 피드백을 제공하세요. - 향후 분석 정확도를 높이려면 예 또는 아니요를 선택합니다.
4. 개요 섹션에서 주요 속성을 검토합니다.
   - 에이전트 신뢰도 - AWS Security Agent가이 조사 결과에서 갖는 신뢰도 수준
   - 심각도 - 색상으로 구분된 배지가 있는 위험 수준
   - 위험 유형 - 보안 위험 범주
   - 검증 상태 - 시뮬레이션된 검증이 활성화되면 실행 중인 환경에서 결과가 악용 가능한 것으로 확인되었는지 여부를 나타냅니다.
     - 확인됨 - 시뮬레이션된 환경에서 취약성이 성공적으로 악용되었습니다.
     - 재현되지 않음 - 대상 런타임에서 취약성을 악용할 수 없음
     - 검증 실패 - 오류로 인해 검증 시도가 결정적이지 않았습니다.
     - 검증되지 않음 -이 결과에 대해 시뮬레이션된 검증이 수행되지 않았습니다. 이는 검증이 활성화되지 않았거나이 결과에 도달하기 전에 검증 단계가 시간 초과된 경우에 발생합니다.
   - 해결됨 - 조사 결과가 해결되었는지 여부(예/아니요)
   - 최종 업데이트 - 최신 업데이트의 타임스탬프
5. 설명 섹션을 확장하여 다음을 읽습니다.

- 보안 문제에 대한 자세한 설명
- 취약성을 악용할 수 있는 방법
- 애플리케이션에 미치는 잠재적 영향
- 에이전트가 결과를 확인하기 위해 수행한 확인 단계

6. 코드 위치 섹션을 확장하여 다음을 확인합니다.
   - 문제가 식별된 특정 파일 경로
   - 각 위치에서 발견된 내용에 대한 간략한 설명
   - 관련 코드에 연결된 행 번호 배지
7. 증거 섹션을 확장하여 검토합니다.
   - 결과를 지원하는 특정 코드, 구성 또는 관측치
   - 각 항목이 취약성을 보여주는 이유에 대한 간단한 설명
   - 각 증거의 영향을 받는 파일 및 행 번호
8. 제안된 수정 사항 섹션을 확장하여 확인합니다.
   - AWS Security Agent가 조사 결과를 해결하기 위해 권장하는 변경 사항
   - 각 권장 변경 사항에 대한 코드 또는 구성의 예

> **Tip**
>
> 코드 위치를 사용하여 리포지토리에서 영향을 받는 파일로 빠르게 이동합니다. 각 위치에는 전체 파일을 읽지 않고도 문제를 이해할 수 있는 충분한 컨텍스트가 포함되어 있습니다.

8단계: 조사 결과 우선순위 지정 및 해결

조사 결과를 검증하고 조치를 취하여 취약성을 해결하고 애플리케이션의 보안 태세를 개선합니다.

에이전트 신뢰도가 높은 심각도가 높은 조사 결과의 경우:

1. 설명 및 코드 위치 섹션을 철저히 검토합니다.
2. 수정 코드를 사용하여 수정 사항이 포함된 풀 요청을 생성하거나 해당 옵션을 활성화한 경우 자동 수정 PRs 검토합니다.
3. 후속 코드 검토 실행을 계획하여 수정 사항이 효과적인지 확인합니다.

에이전트 신뢰도가 높은 중간 심각도 조사 결과의 경우:

1. 위험 허용 범위 및 비즈니스 컨텍스트에 따라 우선순위를 지정합니다.
2. 정기적인 개발 스프린트 계획에 문제 해결 작업을 포함합니다.
3. 여러 중간 심각도 조사 결과가 함께 더 높은 위험을 초래하는지 고려합니다.

중간 또는 낮은 신뢰도 조사 결과의 경우:

1. 설명 및 코드 위치 섹션을 검토하여 취약성이 발생하는 가정과 조건을 검증합니다.
2. 취약성이 유효한 경우 심각도 및 위험 허용 범위에 따라 우선순위를 지정하고 문제 해결 작업을 고려합니다.

9단계: 문제 해결 진행 상황 추적

조사 결과 인터페이스를 사용하고 다시 실행하여 해결된 취약성을 추적합니다.

1. 수정 사항을 구현할 때 결과 해결을 사용하여 조사 결과를 해결된 것으로 표시합니다.
2. 새 코드 검토를 실행하여 수정으로 인해 조사 결과가 해결되고 새 문제가 발생하지 않는지 확인합니다.
3. 코드 검토 세부 정보 페이지의 모든 실행 테이블을 검토하여 시간 경과에 따른 실행 결과를 비교합니다.

> **Tip**
>
> 문제 해결 풀 요청을 병합한 후 동일한 코드 검토의 새 실행을 시작하여 취약성이 해결되었는지 확인합니다. 실행 간 조사 결과 수를 비교하여 진행 상황을 추적합니다.

다음 단계

코드 검토 결과를 검토한 후:

- 즉각적인 해결을 위해 높은 신뢰도로 심각도가 높은 조사 결과의 우선 순위 지정
- 문제 해결 코드를 사용하여 GitHub 소스 조사 결과에 대한 자동 수정 생성(참조the section called “코드 검토 결과 수정”)
- 수정 사항을 구현한 후 추가 코드 검토를 실행하여 문제 해결 확인
- 코드베이스가 발전함에 따라 코드 검토 소스 또는 설정을 조정합니다(참조the section called “코드 검토 활성화”).

코드 검토 결과 수정

코드 검토의 보안 결과를 검토한 후 AWS Security Agent를 사용하여 GitHub 리포지토리 소스의 결과에 대한 코드 수정을 생성할 수 있습니다. 프라이빗 리포지토리의 경우 AWS Security Agent는 제안된 수정 사항이 포함된 풀 요청을 엽니다. 퍼블릭 리포지토리의 경우 AWS Security Agent는 로컬에서 적용할 수 있는 결과에 다운로드 가능한 차이를 연결합니다. S3 소스의 결과는 수동으로 수정해야 합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 조사 결과가 포함된 완료된 코드 검토 실행
- 코드 검토를 위한 소스로 연결된 GitHub 리포지토리
- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 애플리케이션의 아키텍처 및 보안 요구 사항에 대한 지식

코드 수정 작동 방식

결과에 대한 코드 수정을 트리거하면 AWS Security Agent는 결과와 해당 코드 위치를 분석한 다음 코드 수정을 생성합니다. 수정 사항이 전달되는 방법은 소스에 따라 다릅니다.

- 프라이빗 GitHub 리포지토리 - AWS Security Agent는 관련 리포지토리에 풀 요청을 제출합니다. 풀 요청에는 보안 문제 및 변경 사항에 대한 설명이 포함됩니다.
- 퍼블릭 GitHub 리포지토리 - AWS Security Agent는 취약성이 수정되기 전에 공개되지 않도록 제안된 차이를 결과에 연결합니다. 웹 애플리케이션에서 diff를 다운로드하여를 사용하여 로컬로 적용할 수 있습니다`git apply /path/to/code_remediation_changes.diff`.
- S3 소스 - 코드 수정을 사용할 수 없습니다. 결과의 설명, 코드 위치 및 위험 추론을 사용하여 수정 사항을 수동으로 적용합니다.

> **Important**
>
> AWS Security Agent에서 생성한 풀 요청은 리포지토리에 대한 읽기 액세스 권한이 있는 모든 사용자에게 표시됩니다. 병합하기 전에 변경 사항을 검토하여 애플리케이션의 요구 사항에 맞는지 확인합니다.

자동 코드 수정

코드 검토를 생성할 때 자동 코드 수정을 활성화한 경우 AWS Security Agent는 검토가 완료되는 즉시 모든 적격 결과에 대한 수정 사항을 생성합니다. 추가 조치를 취할 필요가 없습니다. 풀 요청은 연결된 프라이빗 GitHub 리포지토리에 나타나고 실행이 완료되면 퍼블릭 GitHub 리포지토리의 결과에 차이가 나타납니다.

수동 코드 수정

자동 코드 수정이 비활성화된 경우 GitHub 소스의 개별 결과에 대한 수정을 트리거할 수 있습니다.

1. 완료된 코드 검토 실행의 결과 탭으로 이동합니다.
2. 해결하려는 결과를 선택합니다.
3. 결과 세부 정보 패널에서 코드 수정을 선택합니다.
4. AWS Security Agent는 코드 수정을 생성하고 리포지토리에 풀 요청을 제출하거나 리포지토리의 가시성에 따라 결과에 다운로드 가능한 차이를 연결합니다.

문제 해결 풀 요청 검토

AWS Security Agent가 프라이빗 GitHub 리포지토리에 문제 해결 풀 요청을 제출한 후:

1. GitHub 리포지토리로 이동합니다.
2. AWS Security Agent에서 생성한 풀 요청을 찾습니다.
3. 다음을 포함한 변경 사항을 검토합니다.
   - 보안 조사 결과 및 수정 사항을 설명하는 설명
   - 취약성을 해결하는 코드가 변경됩니다.
   - 문제 해결 접근 방식에 대한 모든 관련 컨텍스트
4. 수정이 적절한 경우 풀 요청을 병합하거나 종료하고 대체 솔루션을 구현합니다.

> **Tip**
>
> 문제 해결 풀 요청을 병합한 후 새 코드 검토 실행을 시작하여 수정 사항이 조사 결과를 해결하고 새 보안 문제를 일으키지 않는지 확인합니다.

문제 해결 차이 적용

퍼블릭 GitHub 리포지토리의 조사 결과의 경우 다운로드 가능한 diff를 로컬로 적용합니다.

1. 조사 결과 세부 정보 패널의 코드 수정 섹션에서 차이를 다운로드합니다.
2. 리포지토리의 로컬 복제본에서를 실행합니다`git apply /path/to/code_remediation_changes.diff`.
3. 적용된 변경 사항을 검토하고 애플리케이션에 대해 테스트한 다음 일반적인 개발 워크플로를 통해 커밋합니다.

제한 사항

- 코드 수정은 GitHub 리포지토리 소스의 결과에만 사용할 수 있습니다. S3 소스의 결과는 수동으로 수정해야 합니다.
- AWS Security Agent는 취약성 분석을 기반으로 수정 사항을 생성합니다. 병합하거나 커밋하기 전에 모든 변경 사항을 검토하여 애플리케이션에 적합한지 확인합니다.
- 일부 복잡한 조사 결과에는 자동 수정 이외의 수동 개입이 필요할 수 있습니다.

다음 단계

조사 결과를 수정한 후:

- GitHub 리포지토리에서 풀 요청을 검토 및 병합하거나 일반 워크플로를 통해 적용된 차이 커밋
- 새 코드 검토를 실행하여 수정 사항을 확인하고 나머지 문제가 있는지 확인합니다.
- 문제 해결을 확인한 후 웹 애플리케이션의 조사 결과 해결
- 필요에 따라 코드 검토 소스 또는 설정을 조정합니다(참조the section called “코드 검토 활성화”).

S3를 사용하여 차등 코드 스캔 실행

전체 리포지토리 스캔을 수행하는 대신 차등(diff) 코드 스캔을 실행하여 소스 코드에서 변경된 줄만 분석합니다. 차등 스캔은 전체 스캔보다 빠르며 특정 코드 변경을 대상으로 하는 결과를 생성하므로 개발 워크플로에서 사전 병합 검증에 적합합니다.

타사 소스 제어 공급자 이벤트(참조the section called “풀 요청에서 코드 보안 조사 결과 검토”)에 의해 자동으로 트리거되는 풀 요청 기반 코드 검토와 달리 차등 스캔은 AWS Security Agent API 또는 SDK

를 통해 프로그래밍 방식으로 시작됩니다. 통합 diff 파일을 S3에 업로드하고 코드 검토 작업을 시작할 때 참조합니다.

차등 스캔 작동 방식

차등 스캔은 리포지토리의 전체 컨텍스트에서 코드 변경 사항을 분석합니다. 소스 코드가 S3에 업로드된 상태에서 코드 검토 리소스를 생성하면 차등 스캔은 전체 리포지토리를 컨텍스트로 사용하는 동시에 특히 diff에서 변경된 줄에 조사 결과를 집중합니다. 이를 통해 스캔은 인증 흐름 중단, 모듈 간 안전하지 않은 데이터 처리, 기존 취약성을 노출하는 변경 등 변경 사항이 기존 코드와 상호 작용하는 방식에서 발생하는 보안 문제를 식별할 수 있습니다.

스캔 프로세스: . Agent Space에 연결된 S3 버킷에 통합 diff 파일(의 출력`git diff`)을 업로드합니다. diff의 S3 위치를 가리키는 `diffSource` 파라미터를 사용하여 `StartCodeReviewJob` API를 호출합니다. AWS Security Agent는 diff의 변경된 코드만 분석하여 보안 취약성과 조직의 보안 요구 사항 준수를 확인합니다. 결과는 심각도 등급, 코드 위치 및 문제 해결 지침과 함께 변경된 줄로 범위가 지정됩니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 코드 검토가 활성화된 에이전트 공간(참조the section called “코드 검토 활성화”)
- 대상 리포지토리에 대해 이미 생성된 코드 검토 리소스로, 전체 소스 코드가 에이전트 스페이스에 연결된 S3 버킷에 업로드됩니다. 전체 리포지토리는 변경 사항이 기존 코드와 상호 작용하는 방식을 심층적으로 분석할 수 있는 컨텍스트를 제공합니다. 코드 검토 생성 및 소스 코드 업로드에 대한 지침은 the section called “코드 검토 생성” 섹션을 참조하세요.
- 에이전트 스페이스에 연결된 S3 버킷
- S3 버킷에 업로드하고 호출할 수 있는 IAM 권한 `securityagent:StartCodeReviewJob`
- 코드 변경에서 생성된 통합 diff 파일(예:의 출력`git diff main..feature-branch`)

> **Tip**
>
> 가장 정확한 결과를 얻으려면 코드 검토 리소스에 전체 소스 코드의 최신 버전이 S3에 업로드되어 있는지 확인하세요. diff 스캔은이 전체 리포지토리를 컨텍스트로 사용하여 변경된 라인을 분석할 때 애플리케이션 아키텍처, 데이터 흐름 및 기존 보안 제어를 이해합니다.

1단계: diff 파일 생성

스캔하려는 코드 변경 사항을 나타내는 통합 diff를 생성합니다.

```
git diff main..feature-branch > changes.diff
```

또는 스테이징된 변경 사항에 대한 차이를 생성합니다.

```
git diff --cached > changes.diff
```

> **Tip**
>
> diff 파일은 표준 통합 diff 형식이어야 합니다. AWS Security Agent는 파일 경로를 구문 분석하고 diff 헤더에서 줄을 변경하여 분석 범위를 지정합니다.

2단계: S3에 차이 업로드

에이전트 스페이스에 연결된 S3 버킷에 diff 파일을 업로드합니다.

```
aws s3 cp changes.diff s3://my-security-agent-bucket/diffs/changes.diff
```

> **Important**
>
> S3 버킷은 에이전트 스페이스에 이미 연결된 버킷이어야 합니다. 에이전트 스페이스와 연결된 IAM 서비스 역할에는이 위치에 대한 읽기 액세스 권한이 있어야 합니다. S3 버킷 연결에 대한 지침은 the section called “코드 검토 활성화” 섹션을 참조하세요.

3단계: 차등 코드 검토 작업 시작

`diffSource` 파라미터로 `StartCodeReviewJob` API를 호출하여 차등 스캔을 시작합니다.

AWS CLI 사용

```
aws securityagent start-code-review-job \
  --agent-space-id "your-agent-space-id" \
```

```
  --code-review-id "your-code-review-id" \
  --diff-source '{"s3Uri": "s3://my-security-agent-bucket/diffs/changes.diff"}'
```

AWS SDK(Python) 사용

```
import boto3

client = boto3.client('securityagent')

response = client.start_code_review_job(
    agentSpaceId='your-agent-space-id',
    codeReviewId='your-code-review-id',
    diffSource={
        's3Uri': 's3://my-security-agent-bucket/diffs/changes.diff'
    }
)

print(f"Job started: {response['codeReviewJobId']}")
```

AWS SDK(JavaScript/TypeScript) 사용

```
import { SecurityAgentClient, StartCodeReviewJobCommand } from '@aws-sdk/client-
securityagent';

const client = new SecurityAgentClient({ region: 'us-east-1' });

const response = await client.send(new StartCodeReviewJobCommand({
  agentSpaceId: 'your-agent-space-id',
  codeReviewId: 'your-code-review-id',
  diffSource: {
    s3Uri: 's3://my-security-agent-bucket/diffs/changes.diff',
  },
}));

console.log(`Job started: ${response.codeReviewJobId}`);
```

API는 `status`의 `codeReviewJobId` 및와 함께 즉시 반환됩니다`IN_PROGRESS`.

4단계: 스캔 모니터링

코드 검토 작업 상태가 완료될 때까지 폴링합니다.

```
aws securityagent get-code-review-job \
  --agent-space-id "your-agent-space-id" \
  --code-review-id "your-code-review-id" \
  --code-review-job-id "the-job-id"
```

작업은 전체 코드 검토와 동일한 단계인 사전 처리 → 정적 분석 → 완료로 진행됩니다. 차등 스캔은 일반적으로 전체 스캔보다 더 빠르게 완료됩니다. 코드 줄을 더 적게 분석하기 때문입니다.

5단계: 조사 결과 검토

작업이 완료되면 코드 검토 작업에 대한 결과를 나열합니다.

```
aws securityagent list-findings \
  --agent-space-id "your-agent-space-id" \
  --code-review-id "your-code-review-id" \
  --code-review-job-id "the-job-id"
```

각 조사 결과에는 다음이 포함됩니다.

- 보안 문제에 대한 설명
- 심각도 수준(심각, 높음, 중간 또는 낮음)
- diff의 특정 줄을 참조하는 코드 위치
- 잠재적 영향을 설명하는 위험 추론
- 수정 지침

조사 결과를 이해하고 이에 대한 조치를 취하는 방법에 대한 자세한 내용은 섹션을 참조하세요the section called “코드 검토의 결과 검토”.

할당량 및 제한

- diff의 최대 변경된 파일 수: 파일 3,000개 또는 총 diff 크기 5MB
- 결과는 심각도에 따라 우선순위가 지정된 스캔당 30개로 제한됩니다(심각 > 높음 > 중간 > 낮음).

다음 단계

차등 스캔을 실행한 후:

- 조사 결과 검토 및 문제 해결 지침 적용(참조the section called “코드 검토의 결과 검토”)
- 모든 풀 요청에 대한 자동 코드 검토를 위해 풀 요청 설명 활성화(참조the section called “GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화”)
- 정기적으로 전체 코드 검토를 실행하여 개별 변경 사항 이외의 문제를 파악합니다(참조the section called “코드 검토 생성”).
- IDE 통합을 사용하여 개발 환경에서 직접 차등 스캔을 실행합니다(참조the section called “IDE에서 코드 보안 스캔 실행”).

IDE에서 코드 보안 스캔 실행

Kiro 또는 Claude 코드를 사용하여 IDE에서 AWS Security Agent 코드 보안 스캔을 직접 실행합니다. IDE 통합을 사용하면 개발 환경을 벗어나지 않고도 로컬 소스 코드에서 보안 취약성을 스캔하고, 변경된 코드에 대해서만 차등 스캔을 실행하고, 설계 문서에 대한 위협 모델 검토를 수행할 수 있습니다. 조사 결과는 문제 해결 지침과 함께 코드와 함께 표시되며 IDE에서 직접 자동 수정을 적용할 수 있습니다.

IDE 통합 작동 방식

IDE 통합은 AWS Security Agent MCP(모델 컨텍스트 프로토콜) 서버를 사용하여 IDE를 AWS Security Agent 서비스에 연결합니다.

1. MCP 서버는 소스 코드를 ZIP 아카이브로 패키징합니다.
2. 아카이브는 서버가 AWS 계정에서 자동으로 프로비저닝하는 S3 버킷에 업로드됩니다.
3. 서버는 AWS Security Agent API를 호출하여 스캔을 시작합니다.
4. AWS Security Agent는 코드에서 보안 취약성 및 조직의 보안 요구 사항 준수를 분석합니다.
5. 결과는 코드 위치, 설명, 심각도 등급 및 문제 해결 제안과 함께 IDE로 반환됩니다.

MCP 서버는 에이전트 공간 프로비저닝, S3 버킷 생성, IAM 역할 설정, 코드 패키징, 스캔 폴링 등 모든 오케스트레이션을 처리하므로 로컬로 구성된 AWS 자격 증명만 필요합니다.

> **Note**
>
> 유일한 사전 조건은 로컬 환경에 구성된 AWS 자격 증명입니다(예: , `aws configure`AWS SSO 또는 환경 변수를 통해). MCP 서버는 처음 사용할 때 에이전트 공간, IAM 서비스 역할 및 S3 버킷을 자동으로 프로비저닝합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 다음 권한으로 로컬에서 구성된 AWS 자격 증명:
  - `iam:CreateRole`, `iam:PutRolePolicy` (서비스 역할의 일회성 설정용)
  - `s3:CreateBucket`, `s3:PutObject`, `s3:PutPublicAccessBlock`, `s3:PutLifecycleConfiguration`
  - `securityagent:CreateAgentSpace`, `securityagent:UpdateAgentSpace`, `securityagent:ListAgentSpaces`, `securityagent:BatchGetAgentSpaces`
  - `securityagent:CreateCodeReview`, `securityagent:StartCodeReviewJob`, `securityagent:BatchGetCodeReviewJobs`
  - `securityagent:ListFindings`, `securityagent:BatchGetFindings`
  - `securityagent:StartCodeRemediation` (자동 수정의 경우)
  - `sts:GetCallerIdentity`
- uv 설치됨(Python 패키지 실행기)
- Python 3.10 이상
- 다음 IDEs 중 하나:
  - Kiro(AWS Security Agent Power 설치)
  - Claude Code(AWS Security Agent 플러그인 설치)

MCP 서버 설치

Kiro

Kiro Marketplace에서 AWS Security Agent Power를 설치합니다. Power는 자동 보안 스캔 제안을 위해 MCP 서버 구성, 조향 지침 및 수명 주기 후크를 번들링합니다.

또는 프로젝트의에서 MCP 서버를 수동으로 구성합니다. `.kiro/mcp.json`

```
{
  "mcpServers": {
    "security-agent": {
      "command": "uvx",
      "args": ["awslabs.security-agent-mcp-server@latest"],
```

```
      "env": {
        "AWS_PROFILE": "default",
        "AWS_REGION": "us-east-1",
        "FASTMCP_LOG_LEVEL": "ERROR"
      }
    }
  }
}
```

Claude Code

설정, 전체 스캔, diff 스캔, 위협 모델 검토 및 문제 해결에 대한 기술을 제공하는 AWS Security Agent 플러그인을 설치합니다.

프로젝트의에서 MCP 서버를 구성합니다. `.mcp.json`

```
{
  "mcpServers": {
    "awslabs.security-agent-mcp-server": {
      "command": "uvx",
      "args": ["awslabs.security-agent-mcp-server@latest"],
      "env": {
        "AWS_PROFILE": "default",
        "AWS_REGION": "us-east-1",
        "FASTMCP_LOG_LEVEL": "ERROR"
      }
    }
  }
}
```

> **Tip**
>
> Python을 로컬에 설치하지 않으려면 Docker를 통해 MCP 서버를 실행할 수도 있습니다. Docker 구성에 대한 MCP 서버 설명서를 참조하세요.

최초 설정

MCP 서버는 처음 사용할 때 필요한 AWS 리소스를 자동으로 프로비저닝합니다.

Resource	이름 규칙	용도
에이전트 공간	사용자가 선택하거나 자동으로 생성됨	스캔, 검토 및 펜테스트용 컨테이너
IAM 서비스 역할	`SecurityAgentScanRole`	업로드된 코드를 읽기 `securityagent.amazonaws.com` 위해에서 수임
S3 버킷	`security-agent-scans-{account}-{region}`	압축된 소스 코드 저장(30일 자동 만료 수명 주기)

설정을 명시적으로 트리거하려면 IDE에 문의하십시오.

```
Set up the security agent
```

설정은 AWS 자격 증명을 확인하고, 에이전트 스페이스를 생성 또는 재사용하고, IAM 역할 및 S3 버킷을 프로비저닝하고, 준비 상태를 확인합니다. AWS 관리 콘솔에 에이전트 스페이스가 이미 있는 경우 설정이 에이전트 스페이스를 표시하고 사용할 에이전트 스페이스를 묻습니다.

> **Note**
>
> 아직 구성되지 않은 경우 첫 번째 스캔 시 설정이 자동으로 실행됩니다. 별도로 실행할 필요가 없습니다.

전체 보안 스캔 실행

전체 프로젝트에서 보안 취약성을 스캔합니다.

```
Scan this project for security issues
```

IDE:

1. 소스 코드 아카이브(`.git`, `node_modules`, 빌드 아티팩트 및 기타 필수적이지 않은 디렉터리 제외)

2. 아카이브를 S3에 업로드합니다.
3. 전체 코드 검토 작업을 시작합니다.
4. 완료를 위한 폴링(일반적으로 약 1시간)
5. 심각도별로 그룹화된 조사 결과를 제시합니다.

스캔 진행 상황

전체 스캔은 일반적으로 코드베이스 크기에 따라 약 1시간이 걸립니다. IDE는 5분마다 진행 상황을 확인하고 결과가 준비되면 알려줍니다. 언제든지 상태를 요청할 수 있습니다.

```
How's the scan going?
```

차등 스캔 실행

개발 중에 더 빠른 피드백을 얻으려면 git 참조 이후 변경된 코드만 스캔하세요.

```
Scan my changes against main for security issues
```

기본적으로 기본 참조를 지정하지 않으면 스캔은 커밋되지 않은 변경 사항을와 비교합니다`HEAD`. 다른 기본을 명시적으로 지정할 수 있습니다.

```
Diff scan my uncommitted changes
```

차등 스캔은 전체 리포지토리 컨텍스트와 git diff 패치를 모두 업로드한 다음 변경된 줄에만 초점을 맞춘 분석을 실행합니다. 결과는 일반적으로 5~15분 후에 도착합니다.

S3 diff 스캔 API에 대한 자세한 내용은 섹션을 참조하세요the section called “S3를 사용하여 차등 코드 스캔 실행”.

위협 모델 검토 실행

STRIDE 방법론을 사용하여 보안 상황 변경에 대한 설계 문서를 분석합니다.

```
Run a threat model review on my spec
```

IDE는 `requirements.md` 및 `design.md` 파일(일반적으로 아래`.kiro/specs/`)을 식별하고, 소스 코드와 함께 업로드하고, 위협 모델 분석을 실행합니다. 결과는 다음을 식별합니다.

- STRIDE로 분류된 보안 위협(스푸핑, 변조, 거부, 정보 공개, 서비스 거부, 권한 상승)
- 각 위협의 심각도 등급
- 코드베이스의 특정 자산에 미치는 영향
- 완화를 위한 권장 사항

> **Tip**
>
> 사양에서 구현 작업을 생성하기 전에 위협 모델 검토를 실행합니다. 이렇게 하면 코드가 작성되기 전에 보안 설계 문제를 포착할 수 있습니다.

조사 결과 검토

스캔이 완료되면 조사 결과가 심각도별로 그룹화된 IDE에 나타납니다.

```
# CRITICAL: SQL Injection in user-service.ts
   File: src/api/user-service.ts:45
   User input flows directly into SQL query without parameterization

# HIGH: Hardcoded credentials in config.ts
   File: src/config/database.ts:12
   Database password stored in source code
```

위험 유형, 신뢰도 점수, 코드 위치, 문제 해결 코드 등 각 결과에 대한 전체 정보와 `.security-agent/findings-{scan_id}.md` 함께 세부 보고서도에 작성됩니다.

이전 스캔의 결과를 보려면:

```
Show my security findings
```

자동 수정 사항 적용

조사 결과를 검토한 후 자동 문제 해결을 적용합니다.

```
Fix the security findings
```

IDE는 심각도(심각 → 높음 → 중간 → 낮음)별로 조사 결과를 하향 처리하여 각 조사 결과의 수정 지침을 사용하여 코드 수정을 적용합니다. 모든 수정 사항이 적용되면 자동으로 확인 diff 스캔을 실행하여 문제가 해결되었는지 확인합니다.

개별 조사 결과를 수정할 수도 있습니다.

```
Fix the SQL injection in user-service.ts
```

> **Important**
>
> 커밋하기 전에 항상 자동 수정 사항을 검토하세요. 수정으로 인해 기존 기능이 손상되거나 회귀가 발생하지 않는지 확인합니다.

자동 스캔 제안(Kiro)

Kiro Power를 사용하는 경우 AWS Security Agent는 보안에 민감한 코드를 변경한 후 diff 스캔을 자동으로 제안할 수 있습니다. Power는 완료된 각 코딩 턴을 평가하고 변경 사항이 다음에 영향을 미치는 경우 스캔을 제안하는 후크를 설치합니다.

- 인증 또는 권한 부여 로직
- 암호화 또는 보안 암호 처리
- 입력 검증 또는 데이터 삭제
- 네트워크 요청 또는 외부 통합
- 보안 구성(CORS, 헤더, 세션 처리)

를 삭제하여 언제든지 자동 제안을 옵트아웃할 수 있습니다`.kiro/hooks/security-diff-scan-suggester.kiro.hook`.

사용 가능한 스캔 유형

스캔 유형	지속 시간	사용 사례:	명령
전체 스캔	약 1시간	전체 코드베이스에 대한 포괄적인 검토	“프로젝트에서 보안 문제 스캔”

스캔 유형	지속 시간	사용 사례:	명령
Diff 스캔	5~15분	코드만 변경됨(사전 커밋, 사전 PR)	"변경 사항 스캔" 또는 "기본 스캔"
위협 모델	5~30분	디자인 문서 (requirements.md, design.md)	“내 사양에서 위협 모델 검토 실행”

환경 변수

변수	설명	기본값
`AWS_REGION`	SecurityAgent API 호출을 위한 AWS 리전	`us-east-1`
`AWS_PROFILE`	AWS 자격 증명 프로필 이름	기본 프로필
`FASTMCP_LOG_LEVEL`	MCP 서버 로그 수준(디버그, 정보, 경고, 오류)	`WARNING`

문제 해결

"구성되지 않았습니다. 먼저 설정을 실행합니다.”

`.security-agent/config.json`이 없거나 에이전트 스페이스가 더 이상 존재하지 않습니다. IDE에 설정을 실행하도록 요청합니다.

```
Set up the security agent
```

AccessDenied로 스캔 실패

AWS 자격 증명에 사전 조건 섹션에 나열된 필수 IAM 권한이 있는지 확인합니다. 가장 일반적인 누락된 권한은 입니다`iam:CreateRole`(처음 설정에만 필요).

스캔 시간이 초과되거나 너무 오래 걸림

- 대규모 코드베이스에서 전체 스캔은 1시간 이상 걸릴 수 있습니다.
- 반복 개발을 위해 차등 스캔 사용 - 몇 분 안에 완료
- 소스 아카이브에 불필요한 디렉터리가 포함되어 있지 않은지 확인합니다(MCP 서버는 공통 패턴을 자동으로 제외함).

비어 있는 차이 - 스캔 변경 없음

커밋되지 않은 변경 없이 diff 스캔을 실행하면 MCP 서버는 "변경 없음 vs HEAD"를 보고하고 스캔을 시작하지 않습니다. 먼저 변경 사항을 커밋 또는 스테이징하거나 다른 기본 참조를 지정합니다.

할당량 및 제한

- 최대 소스 아카이브 크기: 2GB
- S3 버킷 수명 주기: 30일 후 업로드된 코드 자동 삭제

다음 단계

첫 번째 IDE 보안 스캔을 실행한 후:

- 자동 GitHub 통합에 대한 풀 요청 코드 검토 설명 활성화(참조the section called “GitHub 리포지토리에 대한 풀 요청 코드 검토 활성화”)
- 조직별 정책 검증을 위한 보안 요구 사항 구성(참조the section called “보안 요구 사항 관리”)
- 정기적인 전체 스캔을 실행하여 전체 코드베이스에서 문제를 파악합니다(참조the section called “코드 검토 생성”).
- 구현 전에 새로운 기능 사양에 대한 위협 모델 검토 사용

침투 테스트 생성

테스트 범위, 대상 도메인 및 AWS 리소스 액세스를 구성하여 웹 애플리케이션에 대한 자동 침투 테스트를 설정합니다. 침투 테스트는 확인된 도메인에 대한 실제 공격 시나리오를 시뮬레이션하여 애플리케이션 실행의 보안 취약성을 식별하는 데 도움이 됩니다.

AWS Security Agent는 구성된 범위 및 권한을 기반으로 웹 애플리케이션에 대한 포괄적인 보안 테스트를 수행하여 공격자가 발견하기 전에 악용 가능한 취약성에 대한 자세한 결과를 제공합니다.

이 절차에서는 테스트 세부 정보를 구성하고, 테스트 범위를 정의하고, 필요한 권한을 설정하여 침투 테스트를 생성합니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 테스트를 위해 하나 이상의 확인된 도메인
- AWS Security Agent에 대한 적절한 권한이 있는 IAM 역할
- 애플리케이션의 아키텍처 및 중요 경로 이해

침투 테스트 생성 시작

에이전트 웹 앱에서 침투 테스트 생성 페이지로 이동합니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 침투 테스트 섹션으로 이동합니다.
3. 침투 테스트 생성을 선택합니다.

> **Tip**
>
> 확인된 도메인만 침투 테스트에 포함될 수 있습니다. 관리자에게 AWS 관리 콘솔에서 도메인을 확인하도록 요청합니다. the section called “침투 테스트를 위한 애플리케이션 도메인 활성화”을(를) 참조하세요.

침투 테스트 이름 지정

이 침투 테스트의 목적과 범위를 식별하는 데 도움이 되는 설명 이름을 제공합니다.

1. 침투 테스트 이름 필드에 침투 테스트를 설명하는 이름을 입력합니다.

   Example

   이름은 테스트 중인 애플리케이션, 환경 또는 구성 요소를 명확하게 식별해야 합니다. 최대 100자입니다.

침투 테스트 범위 구성

테스트할 도메인 및 URL 경로를 정의하고 테스트 경계를 제어하도록 선택적 제외를 구성합니다.

대상 도메인 추가

보안 취약성을 적극적으로 테스트할 확인된 도메인을 지정합니다.

1. 침투 테스트 범위 섹션에서 대상 URLs 찾습니다.
2. 확인된 도메인 섹션을 확장하여 사용 가능한 도메인을 확인합니다.
3. 대상 URL 필드에 대상 도메인 URL을 입력합니다.

> **Important**
>
> 확인된 도메인만 테스트할 수 있습니다. URL은 AWS Security Agent에서 이전에 확인한 도메인 아래에 있어야 합니다. 확인된 도메인의 하위 도메인은 별도의 확인이 필요하지 않습니다.

4. 여러 대상 도메인을 추가하려면:
   a. 도메인 추가를 선택합니다.
   b. 각 추가 도메인 URL을 입력합니다.
5. 대상 도메인을 제거하려면 도메인 URL 옆의 제거를 선택합니다.

> **Tip**
>
> 최상의 결과를 얻으려면 APIs, 인증 서비스 및 콘텐츠 전송을 위한 하위 도메인을 포함하여 애플리케이션 사용자 흐름의 일부인 모든 도메인을 포함하세요. 확인된 상위 도메인의 하위 도메인은 별도의 확인이 필요하지 않습니다.

위험 유형 제외(선택 사항)

애플리케이션에 적용되지 않는 경우 테스트에서 제외할 특정 위험 범주를 선택합니다.

1. 위험 유형 제외 필드를 찾습니다.
2. 드롭다운을 선택하여 사용 가능한 위험 유형을 확인합니다.
3. 침투 테스트에서 제외할 위험 유형을 하나 이상 선택합니다.

> **Note**
>
> 위험 유형을 제외하면 테스트 범위가 제한됩니다. 애플리케이션과 관련이 없거나 별도로 테스트하려는 위험 유형만 제외합니다.

out-of-scope URL 경로 추가(선택 사항)

침투 테스트 중에 테스트해서는 안 되는 URL 경로를 지정합니다. AWS Security Agent는 지정된 경로와 그 아래에 중첩된 모든 경로를 제외합니다. 예를 들어 범위를 out-of-scope URL`https://example.com/admin`로를 추가하는 경우 `https://example.com/admin/tools`도 out-of-scope.

1. Out-of-scope URLs 섹션을 찾습니다.
2. Out-of-scope URLs 입력 필드에 제외할 URL 경로(예: `/admin/delete` 또는 `/api/reset`)를 입력합니다.

> **Warning**
>
> Out-of-scope 경로는 취약성에 대해 테스트되지 않습니다. 파괴적 작업 또는 민감한 관리 함수와 같이 테스트 중에 액세스해서는 안 되는 경로만 제외해야 합니다.

3. out-of-scope 경로를 여러 개 추가하려면:
   a. URL 추가를 선택합니다.
   b. 각 추가 경로를 입력합니다.
4. 경로를 제거하려면 경로 옆의 제거를 선택합니다.

액세스 가능한 도메인 추가(선택 사항)

테스트에 필요하지만 취약성 테스트의 대상이 아닌 도메인을 지정합니다.

1. 액세스 가능한 URLs 섹션을 찾습니다.
2. 액세스 가능한 URLs 입력 필드에 테스트 중에 액세스할 수 있어야 하는 도메인을 입력합니다.

> **Note**
>
> 대상 도메인 외부에 있는 타사 서비스(예: Okta, Auth0, Stripe)에 대한 액세스 가능한 도메인을 추가합니다. 이는 AWS Security Agent가 테스트 중에 로그인 및 탐색을 위해 이러한

URLs에 액세스할 수 있도록 하기 위해 필요합니다. AWS Security Agent는 이러한 도메인을 침투 테스트하지 않으며 액세스 목적으로만 사용됩니다. 액세스 가능한 도메인은 대상과 다른 도메인에 속해 있더라도 소유권 확인이 필요하지 않습니다. 대상 도메인에만 확인된 소유권이 필요합니다.

3. 액세스 가능한 도메인을 여러 개 추가하려면:
   a. URL 추가를 선택합니다.
   b. 각 추가 도메인을 입력합니다.
4. 도메인을 제거하려면 도메인 옆의 제거를 선택합니다.

사용자 지정 HTTP 헤더 추가(선택 사항)

침투 테스트 중에 AWS Security Agent의 요청에 추가할 사용자 지정 HTTP 헤더를 지정합니다.

1. 사용자 지정 HTTP 헤더 섹션을 찾습니다.
2. 사용자 지정 HTTP 헤더 입력 필드에 아웃바운드 요청과 연결할 헤더 이름과 값을 입력합니다.

> **Note**
>
> 다른 사용자 지정 사용자 에이전트 헤더 값을 지정하지 않는 한 AWS Security Agent는 기본적으로 사용자 에이전트 세트에 대한 사용자 지정 헤더를 securityagent에 추가합니다.

3. 여러 사용자 지정 헤더를 추가하려면:
   a. 헤더 추가(Add header)를 선택합니다.
   b. 각 추가 사용자 지정 헤더를 입력합니다.
4. 사용자 지정 헤더를 제거하려면 헤더 옆의 제거를 선택합니다.

IAM 역할 구성

이 침투 테스트를 위해 미리 구성된 서비스 역할을 선택합니다. AWS Security Agent는 에이전트 스페이스를 설정할 때 관리자가 IAM 역할을 구성하는 에이전트 스페이스 기반 권한 모델을 사용합니다. 이미 구성되어 사용할 준비가 된 역할 중에서 선택합니다.

1. 권한 섹션에서 서비스 역할 드롭다운을 찾습니다.
2. AWS Security Agent에 필요한 AWS 리소스에 대한 액세스 권한을 부여하는 IAM 역할을 선택합니다.

> **Important**
>
> 선택한 IAM 역할에는 VPC 리소스, CloudWatch Logs 및 침투 테스트에 필요한 기타 AWS 서비스에 액세스할 수 있는 권한이 있어야 합니다. 역할이 AWS Security Agent와 올바른 신뢰 관계를 맺고 있는지 확인합니다.

3. CloudWatch 로그 그룹 드롭다운을 찾습니다.
4. 침투 테스트 로그가 저장될 로그 그룹을 선택합니다(선택 사항).

> **Note**
>
> 선택한 CloudWatch 로그 그룹은 요청, 수신된 응답 및 발견된 취약성을 포함하여 침투 테스트 실행에 대한 자세한 로그를 저장합니다.
> 로그 그룹을 선택하지 않으면 새 CloudWatch 로그 그룹이 `/aws/securityagent` 접두사와 함께 자동으로 생성되어 침투 테스트 로그를 저장합니다.

자동 코드 수정

자동 문제 해결 활성화 확인란을 선택합니다.

> **Important**
>
> 소스 코드 리포지토리의 보안 조사 결과를 해결하기 위해 AWS Security Agent는 리포지토리에 풀 요청을 제출할 수 있습니다. 풀 요청은 리포지토리에 대한 읽기 액세스 권한이 있는 모든 사용자에게 표시될 수 있습니다.

VPC 리소스 구성(선택 사항)

대상 도메인이 프라이빗이고 VPC 내에서 호스팅되는 경우 AWS Security Agent가 침투 테스트를 실행해야 하는 VPC 설정을 구성합니다. 이 단계는 공개적으로 액세스할 수 없는 애플리케이션에만 필요합니다.

> **Note**
>
> 대상 도메인에 공개적으로 액세스할 수 있는 경우이 단계를 건너뜁니다. VPC 구성은 Amazon Virtual Private Cloud 내에서 호스팅되는 프라이빗 애플리케이션을 테스트하는 경우에만 필요합니다.

침투 테스트 환경의 VPC, 서브넷 및 보안 그룹을 선택합니다.

1. VPC 섹션에서 VPC ID 드롭다운을 찾습니다.
2. 대상 도메인이 호스팅되는 VPC를 선택합니다.

> **Important**
>
> 선택한 VPC에는 3단계에서 지정한 대상 도메인이 포함되어야 합니다. VPC에 AWS Security Agent가 애플리케이션에 액세스할 수 있도록 허용하는 적절한 라우팅 및 네트워크 구성이 있는지 확인합니다.

3. 서브넷 드롭다운을 찾습니다.
4. 침투 테스트를 실행해야 하는 서브넷을 하나 이상 선택합니다.

> **Note**
>
> 대상 애플리케이션에 대한 네트워크 액세스 권한이 있는 서브넷을 선택합니다. 침투 테스트는 이러한 서브넷에 배포된 리소스에서 실행됩니다.

5. 보안 그룹 드롭다운을 찾습니다.
6. 침투 테스트에 대한 네트워크 액세스를 제어하는 보안 그룹을 선택합니다.

> **Important**
>
> 선택한 보안 그룹은 대상 도메인 및 액세스 가능한 도메인으로의 아웃바운드 트래픽을 허용해야 합니다. 보안 그룹 규칙이 포괄적인 테스트에 필요한 네트워크 액세스를 허용하는지 확인합니다.

인증 자격 증명 구성(선택 사항)

대상 도메인에 인증이 필요한 경우 AWS Security Agent가 침투 테스트 중에 애플리케이션의 보호 영역에 액세스할 수 있도록 자격 증명을 제공합니다. 이 단계는 사용자 인증이 필요한 애플리케이션에만 필요합니다.

> **Note**
>
> 대상 도메인에 인증이 필요하지 않거나 테스트하려는 모든 영역에 공개적으로 액세스할 수 있는 경우이 단계를 건너뜁니다. 애플리케이션의 인증된 섹션을 테스트하기 위해 AWS Security Agent가 필요한 경우에만 자격 증명을 구성합니다.

자격 증명 추가

AWS Security Agent가 애플리케이션에 액세스하는 데 사용할 인증 자격 증명을 제공합니다.

1. 자격 증명 #1 섹션에서 자격 증명 입력 방법을 선택합니다.
   - 자격 증명 입력 - 자격 증명을 AWS Security Agent에 직접 입력합니다.
   - 고급 설정 - 민감한 자격 증명 정보의 경우 AWS Secrets Manager 또는 AWS Lambda 함수와 같은 고급 옵션을 사용합니다. 세부 정보는 the section called “침투 테스트를 위한 인증 자격 증명 제공” 섹션을 참조하세요.

> **Tip**
>
> 프로덕션 환경 또는 민감한 자격 증명의 경우 고급 설정 옵션을 사용하여 AWS Secrets Manager 또는 Systems Manager Parameter Store에 저장된 자격 증명을 안전하게 참조하는 것이 좋습니다.

자격 증명 세부 정보 입력

인증된 계정의 사용자 이름과 암호를 입력합니다.

1. 사용자 이름 필드에 인증할 사용자 이름을 입력합니다.
2. 암호 필드에 인증을 위한 암호를 입력합니다.

> **Important**
>
> 제공한 자격 증명에 테스트하려는 영역에 적합한 액세스 수준이 있는지 확인합니다. 자격 증명은 관리 권한이 아닌 일반적인 사용자의 액세스 수준을 나타내야 합니다.

액세스 도메인 선택

인증에 이러한 자격 증명을 사용할 대상 도메인을 지정합니다.

1. 액세스 도메인 드롭다운에서 이러한 자격 증명을 사용할 도메인을 선택합니다.

> **Note**
>
> 다른 자격 증명이 필요한 대상 도메인이 여러 개 있는 경우이 자격 증명 구성을 완료한 후 다른 자격 증명 추가를 클릭하여 자격 증명 세트를 추가할 수 있습니다.

에이전트 로그인 프롬프트 구성(선택 사항)

애플리케이션의 인증 프로세스를 통해 AWS Security Agent를 안내하는 지침을 제공합니다.

1. 인증 흐름에 특정 지침이 필요한 경우 에이전트 로그인 프롬프트 섹션을 확장합니다.
2. 애플리케이션의 로그인 흐름에서 제공된 자격 증명을 사용하는 방법을 설명하는 지침을 입력합니다.

> **Note**
>
> 에이전트 로그인 프롬프트는 자격 증명을 애플리케이션에 적용하는 방법을 에이전트에게 알려줍니다. 이는 복잡한 인증 흐름, 다단계 로그인 프로세스 또는 비표준 로그인 절차가 있는 애플리케이션에 유용합니다. "/로그인으로 이동, '이메일' 필드에 사용자 이름 입력, 암호 입력, '로그인' 선택"과 같은 step-by-step 지침을 포함합니다.

여러 자격 증명 추가(선택 사항)

애플리케이션에 여러 자격 증명 세트가 필요하거나 다른 도메인에 별도의 인증이 필요한 경우 자격 증명 세트를 추가합니다.

1. 첫 번째 자격 증명 구성을 완료한 후 다른 자격 증명 추가를 선택합니다.
2. 각 추가 자격 증명 세트에 대해 자격 증명 구성 단계를 반복합니다.
3. 자격 증명 세트를 제거하려면 자격 증명 헤더 옆의 제거를 선택합니다.

> **Tip**
>
> 다양한 사용자 역할을 테스트하거나, 여러 인증된 도메인에 액세스하거나, 애플리케이션에서 역할 기반 액세스 제어를 확인할 때 여러 자격 증명을 구성합니다.

추가 리소스 연결(선택 사항)

AWS Security Agent가 보다 철저하고 정확한 침투 테스트를 수행할 수 있도록 보조 리소스를 제공합니다. 추가 리소스에는 애플리케이션에 대한 컨텍스트를 제공하는 아키텍처 다이어그램, API 설명서, 구성 파일, GitHub 리포지토리 또는 S3-hosted 자료가 포함될 수 있습니다.

> **Note**
>
> 추가 리소스는 선택 사항이지만 권장됩니다. 애플리케이션에 대한 포괄적인 정보를 제공하면 철저한 테스트 범위를 보장하고, 오탐을 줄이고, 실행 가능한 결과를 얻을 수 있습니다.

침투 테스트에 리소스 추가

기존 리소스를 선택하거나 침투 테스트를 안내하는 데 도움이 되는 새 파일을 업로드합니다.

1. 연결된 리소스 섹션에서 다음을 수행할 수 있습니다.
   - AWS Security Agent에 이미 연결된 리소스(예: GitHub 리포지토리 또는 S3 버킷) 중에서 선택하려면 사용 가능 중에서 선택을 선택합니다.
   - 업로드를 선택하여 로컬 시스템에서 직접 새 파일을 추가합니다.

> **Tip**
>
> 유용한 리소스에는 API 설명서, 아키텍처 다이어그램, OpenAPI/Swagger 사양, 구성 파일, 인증 흐름도 및 애플리케이션의 구조와 동작을 설명하는 기타 자료가 포함됩니다.

사용 가능한 리소스 중에서 선택

AWS Security Agent와 이미 통합된 리소스 중에서 선택합니다.

1. 기존 리소스에서 선택을 선택합니다.
2. 다음과 같이 연결된 소스에서 사용 가능한 리소스 목록을 찾습니다.
   - GitHub 리포지토리 탭 아래의 GitHub 리포지토리
   - S3 버킷
   - 이전에 업로드한 파일
   - 설명서 리포지토리
3. 침투 테스트에 포함할 리소스를 선택합니다.
4. 침투 테스트에 추가를 선택하여 선택한 리소스를 연결합니다.

Example

AWS Security Agent가 애플리케이션 컨텍스트를 이해하고 풀 요청을 통해 ready-to-implement할 수 있는 코드 수정을 생성할 수 있도록 관련 GitHub 리포지토리를 선택하고 Pentest에 추가하는 것이 좋습니다(활성화된 경우).

> **Note**
>
> 사용 가능한 소스에서 선택한 리소스는 원래 위치와 동기화된 상태로 유지됩니다. GitHub 리포지토리 또는 S3 파일을 업데이트하는 경우 침투 테스트는 업데이트된 버전을 사용합니다.

> **Note**
>
> pentest와 연결된 프라이빗 VPC와 GitHub 리포지토리가 구성된 경우 GitHub 리소스를 가져오기 위해 프라이빗 VPC를 통해 GitHub에 액세스할 수 있는지 확인합니다. 대부분의 경우 VPC NAT 게이트웨이를 통한 아웃바운드 트래픽이 기본적으로 허용되는지 확인하거나 GitHub IPs(GitHub Meta API 엔드포인트 참조).

새 리소스 업로드

로컬 시스템에서 직접 파일을 업로드하거나 AWS Security Agent에 일반 텍스트 콘텐츠를 제공합니다.

1. 업로드를 선택합니다.
2. 다음 입력 방법 중 하나를 선택합니다.
   - 로컬 파일 업로드 - 로컬 시스템에서 파일을 하나 이상 선택합니다.
   - 일반 텍스트 붙여넣기 - 텍스트 콘텐츠를 입력 필드에 직접 입력하거나 붙여넣습니다. 업로드를 선택합니다.
3. 그런 다음 추가를 선택하여 업로드를 완료합니다.
4. 업로드된 리소스는 연결된 리소스 테이블에 나타납니다.

> **Tip**
>
> 별도의 파일을 생성하지 않고 API 엔드포인트 목록, URL 패턴, 테스트 지침 또는 기타 텍스트 기반 정보를 빠르게 제공하려는 경우 일반 텍스트 옵션을 사용합니다.

> **Important**
>
> 업로드된 파일 및 붙여넣은 콘텐츠에 프로덕션 자격 증명, 프라이빗 키 또는 개인 식별 정보(PII)와 같은 민감한 정보가 포함되어 있지 않은지 확인합니다. 삭제된 버전의 구성 파일 및 설명서를 사용합니다.

기존 리소스 연결

기존 리소스는 이전에 AWS Security Agent에 업로드한 리소스, S3 버킷 및 통합 GitHub 리포지토리에서 가져올 수 있습니다. 기존 리소스에서 선택을 선택하여 선택합니다.

연결된 리소스 관리

침투 테스트에 연결된 리소스를 검토, 구성 및 제거합니다.

연결된 리소스 테이블에는 침투 테스트에 포함된 모든 리소스가 다음 정보와 함께 표시됩니다.

- 이름 - 파일 이름 또는 리소스 식별자
- 유형 - 리소스 범주(업로드된 파일, S3 리소스, GitHub 리포지토리 등)

리소스를 관리하려면:

1. 확인란을 사용하여 리소스를 하나 이상 선택합니다.
2. 침투 테스트에서 제거를 선택하여 선택한 리소스를 분리합니다.

> **Note**
>
> 열 헤더를 클릭하여 이름 또는 유형별로 테이블을 정렬할 수 있습니다. 이렇게 하면 여러 파일로 작업할 때 리소스를 구성하는 데 도움이 됩니다.

침투 테스트 생성

침투 테스트 구성을 완료하고 시작합니다.

모든 설정을 구성한 후 침투 테스트를 생성할 준비가 되었습니다.

1. 모든 구성 섹션을 검토하여 정확성을 확인합니다.
2. 다음 옵션 중 하나를 선택하세요.
   - 즉시 실행하지 않고 구성을 저장하려면 침투 생성을 선택합니다.
   - 생성 및 실행을 선택하여 구성을 저장하고 즉시 침투 테스트를 시작합니다.
   - 침투 테스트 구성을 취소하려면 취소를 선택합니다.

> **Important**
>
> 침투 테스트를 실행하기 전에 다음을 확인합니다.
>
> - 모든 대상 도메인이 올바르게 확인되고 액세스할 수 있음
> - IAM 역할에 적절한 권한이 있음
> - Out-of-scope 경로가 제대로 구성되어 테스트 중단 작업을 방지합니다.
> - 모든 대상 도메인에서 보안 테스트를 수행할 수 있는 권한이 있음

> **Note**
>
> 침투 테스트가 시작된 후 침투 테스트 실행 섹션에서 진행 상황을 모니터링할 수 있습니다. 테스트를 완료하는 데 몇 시간이 걸릴 수 있습니다. 대부분의 애플리케이션의 범위와 복잡성에 따라 16시간 이내에 완료됩니다.

침투 테스트 결과 검토

AWS Security Agent가 Pentest를 시작한 후 침투 테스트 로그 페이지에서 실시간으로 Pentest 실행을 모니터링합니다. AWS Security Agent는 pentest 중에 모든 작업을 기록합니다. 완료 후 애플리케이션 개요, 식별된 엔드포인트 적용 범위, 보안 조사 결과의 위험 평가가 포함된 Pentest 요약을 검토합니다.

보안 조사 결과를 평가하여 애플리케이션 취약성을 해결합니다. 각 결과에는 영향 평가, 심각도 등급, 지원 증거 및 문제 해결 풀 요청 세부 정보(자동 코드 문제 해결이 활성화된 경우)가 포함됩니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 완료되었거나 진행 중인 침투 테스트 실행
- AWS Security Agent 웹 애플리케이션에 대한 액세스

1단계: 침투 테스트 실행에 액세스

침투 테스트 실행으로 이동하여 개요, 로그 및 결과 페이지를 봅니다.

1. AWS Security Agent 웹 애플리케이션에 로그인합니다.
2. 침투 테스트 섹션으로 이동합니다.
3. 목록에서 검사할 침투 테스트 실행을 선택합니다.

> **Tip**
>
> 침투 테스트 세부 정보 페이지에는 테스트 상태, 완료 날짜 및 식별된 결과 수에 대한 요약이 표시됩니다.

2단계: 테스트 진행 상황 모니터링

단계 표시기를 사용하여 침투 테스트 실행 진행 상황을 추적합니다.

1. 페이지 헤더 아래에서 가로 단계 표시기를 찾습니다.
2. 각 테스트 단계의 상태를 검토합니다.
   - Preflight - 초기 설정 및 연결 검사
   - 정적 분석 - 코드 및 구성 분석
   - Pentests - 런타임 테스트 및 취약성 스캔
   - 마무리 - 최종 검증 및 보고서 생성

> **Note**
>
> 각 단계에는 상태 표시기(완료, 진행 중 또는 보류 중)가 표시됩니다. 결과는 테스트 프로세스 전체에서 검색되고 검증되며, 각 단계가 완료될 때마다 새로운 취약성이 나타납니다.

3단계: 침투 테스트 실행 개요 탭으로 이동

1. 실행 요약 섹션에서는 테스트 상태, 기간 및 기타 상위 수준 세부 정보를 제공합니다. 또한 심각도 수준 및 위험 유형별로 분류된 보안 조사 결과의 대시보드를 제공합니다.
2. AWS Security Agent의 애플리케이션 개요는 침투 테스트 실행의 요약을 제공합니다.
3. AWS Security Agent에서 검색한 엔드포인트는 Pentest 실행 중에 AWS Security Agent에서 검색하고 테스트한 모든 엔드포인트 목록을 제공합니다.

4단계: 침투 테스트 로그 탭으로 이동

Pentest 중에 실행된 AWS Security Agent의 모든 작업에 대한 세부 로그에 액세스합니다.

1. 작업은 작업 유형 및 위험 유형별로 분류됩니다.
2. 세부 로그를 보려면 특정 작업을 선택합니다.
   - 테스트 요약 - 에이전트 작업 및 결과에 대한 상위 수준 요약
   - 침투 테스트 로그 - 모든 테스트 활동의 세부 로그

> **Note**
>
> 검사기 작업은 각 범주의 결과를 검증하는 로그를 제공합니다.

> **Note**
>
> 조사 결과 탭에는 왼쪽에 조사 결과 목록이 있고 오른쪽에 선택한 조사 결과 세부 정보가 있는 분할 보기가 표시됩니다.

5단계: 조사 결과로 이동

목록의 각 결과에는 중요도를 빠르게 평가하는 데 도움이 되는 주요 정보가 표시됩니다.

> **Note**
>
> 기본 신뢰도 필터
>
> 기본적으로 에이전트 신뢰도가 높은 결과만 표시됩니다. 중간 또는 낮은 에이전트 신뢰도와 거짓 긍정으로 조사 결과를 표시하려면 확인되지 않은 조사 결과 숨기기 토글을 끕니다.

각 결과 카드에 표시된 정보를 검토합니다.

- 결과 이름 - 취약성의 제목 및 식별자입니다.
- 신뢰도 배지 - 결과에 대한 에이전트의 신뢰도 수준을 나타냅니다(높음, 중간 또는 낮음).
- 심각도 배지 - 색상 코딩이 있는 위험 수준을 표시합니다.
  - 심각(빨간색) - 즉각적인 조치가 필요합니다. 악용 시 시스템 손상으로 이어질 수 있습니다.
  - 높음(빨간색) - 즉각적인 주의가 필요합니다. 악용 시 상당한 보안 영향이 발생할 수 있습니다.
  - 중간(주황색) - 적절한 기간 내에 해결해야 합니다. 전반적인 보안 위험에 기여합니다.
  - 낮음(노란색) - 정기 유지 관리의 일부로 해결할 수 있으며 즉각적인 위험을 최소화합니다.
  - 정보 제공(파란색) - 정보 제공 목적, 최소 위험 또는 즉각적인 위험 없음
- 마지막 업데이트 타임스탬프 - 조사 결과가 마지막으로 수정되거나 검증된 시기를 표시합니다.
- 설명 미리 보기 - 취약성에 대한 간략한 요약

> **Important**
>
> 심각 또는 높은 심각도 배지와 높은 신뢰도 수준으로 조사 결과의 우선순위를 지정합니다. 이는 즉각적인 수정이 필요한 검증된 취약성을 나타내기 때문입니다.

6단계: 결과 세부 정보 검토

개별 조사 결과를 선택하여 각 취약성에 대한 포괄적인 정보를 봅니다.

1. 왼쪽 패널에서 결과 이름을 선택하여 오른쪽 패널에 세부 정보를 표시합니다.
2. 검증 상태를 검토합니다.

> **Note**
>
> 결과에 알 수 없음 "이 결과는 AWS Security Agent에서 아직 검증되지 않았습니다"가 표시되면 취약성 감지가 여전히 확인 중임을 의미합니다. 이러한 결과는 수동 확인이 필요할 수 있습니다.

3. 상단에 표시된 키 속성을 검토합니다.
   - 에이전트 신뢰도 - AWS Security Agent가이 조사 결과에서 갖는 신뢰도 수준
   - 심각도 - 색상으로 구분된 배지가 있는 위험 수준
   - 로그 찾기 - "작업 및 로그 추적"을 선택하여 자세한 실행 로그 및 증거를 봅니다.
   - 위험 유형 - 보안 위험의 범주 또는 유형(예: Authentication Bypass, SQL Injection)
4. 설명 섹션을 확장하여 다음을 읽습니다.
   - 취약성에 대한 자세한 설명
   - 취약성 작동 방식
   - 보안 위험을 나타내는 이유
   - 애플리케이션에 미치는 잠재적 영향
5. 위험 추론 섹션을 확장하여 심각도 계산을 이해합니다.
   - CVSS(Common Vulnerability Scoring System) 지표 분석
   - 공격 벡터(AV) - 취약성을 악용하는 방법
   - 공격 복잡성(AC) - 악용이 얼마나 어려운지
   - 권한 필요(PR) - 필요한 액세스 수준
   - 사용자 상호 작용(UI) - 사용자 작업이 필요한지 여부

- 범위(S) - 취약성이 다른 구성 요소에 영향을 미치는지 여부
- 기밀성, 무결성 및 가용성에 미치는 영향

6. 재생성 단계 섹션을 확장하여 확인합니다.
   - 취약성을 다시 생성하기 위한 세부 기술 단계
   - 요청 및 응답 예제
   - 문제를 트리거하는 특정 파라미터 또는 조건
7. 확인 스크립트 섹션에서는 결과를 재현할 수 있는 실행 방법을 제공합니다. 이 섹션(사용 가능한 경우)을 확장하여 다음을 확인합니다.
   - 지침 - 확인 스크립트를 설정하고 실행하는 방법
   - 환경 변수 - 스크립트를 실행하기 전에 구성해야 하는 필수 변수를 나열합니다. 보안을 위해 민감한 값이 수정됩니다.
   - 스크립트 다운로드 - 실행 파일 확인 스크립트를 다운로드하도록 선택합니다.

> **Note**
>
> 확인 스크립트는 확인된 취약성에만 사용할 수 있습니다. 에이전트는 실행 파일 복제 스크립트를 성공적으로 생성하고 검증해야 합니다.

> **Note**
>
> 확인 스크립트는 생성형 AI를 사용하여 생성됩니다. 실행 전에 스크립트를 검토하고 테스트 권한이 있는 시스템에 대해서만 스크립트를 실행합니다. AI 생성 스크립트를 책임감 있게 테스트하는 방법에 대한 지침은 AWS 책임 AI 정책을 참조하세요.

> **Tip**
>
> 확인 스크립트는 결과를 독립적으로 재현하는 실행 가능한 방법을 제공합니다. 자체 자격 증명으로 필요한 환경 변수를 설정한 다음 대상 시스템에 대해 스크립트를 실행하여 취약성이 존재하는지 확인합니다.

> **Tip**
>
> "작업 및 로그 추적" 링크를 사용하여 취약성을 보여주는 HTTP 요청, 응답 및 악용 시도를 포함한 전체 증거 패키지에 액세스합니다.

조사 결과 편집

조사 결과를 편집하여 세부 정보를 수정하거나 에이전트의 평가를 구체화할 수 있습니다. 다음 필드를 편집할 수 있습니다.

- name - 조사 결과의 제목입니다.
- 설명 - 보안 취약성에 대한 자세한 설명
- 상태 - 조사 결과의 현재 상태
- riskType - 식별된 보안 위험 유형
- riskLevel - 심각도 수준(CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL)
- riskScore - 숫자 위험 점수
- 추론 - 할당된 위험 점수의 근거
- attackScript Proof-of-concept 코드
- customerNote - 편집 근거를 설명하는 선택적 메모

결과를 편집하려면:

1. 결과 세부 정보 페이지로 이동합니다.
2. 편집 아이콘을 선택하여 이전 목록의 필드를 수정합니다.
3. 변경 내용을 저장합니다.

> **Note**
>
> 편집 내용은 즉시 저장되고 결과에 반영됩니다. 결과 개인화가 활성화된 경우 편집 가능한 모든 필드를 사용하여 향후 실행을 위한 에이전트의 기본 설정을 구체화합니다.

원래 에이전트 버전 보기

결과를 편집하면 결과 세부 정보 헤더에 버전 선택기가 나타납니다. 이렇게 하면 원래 에이전트 평가를 볼 수 있습니다.

1. 결과 세부 정보 헤더에서 버전 선택기를 찾습니다.
2. 원본을 선택하면 에이전트가 원래 생성한 결과를 볼 수 있습니다.
3. 최신을 선택하여 편집 내용이 적용된 현재 버전으로 돌아갑니다.

맞춤형 조사 결과 검토

조사 결과 개인화가 활성화되면 AWS Security Agent는 조사 결과에 대한 편집 내용을 학습합니다. 에이전트는 향후 침투 테스트 실행에서 유사한 결과에 이러한 기본 설정을 적용합니다. 이전 편집에 따라 조사 결과가 조정되면 세부 정보 패널에 개인화 변경 섹션이 표시됩니다. 이 기능 활성화에 대한 자세한 내용은 결과 개인화 활성화 또는 비활성화를 참조하세요.

1. 결과 세부 정보 패널에서 개인화 변경 섹션을 찾습니다.

> **Note**
>
> 개인화 변경 섹션은 AWS Security Agent가 이전 편집에 맞게 조정한 결과에만 표시됩니다. 학습된 기본 설정 없음과 일치하는 결과는이 섹션 없이 에이전트가 생성한 것과 정확히 동일하게 표시됩니다.

2. 변경된 내용과 이유에 대한 요약을 검토합니다. 요약에는 원래 에이전트 생성 값, 개인화된 값 및 추론과 함께 조정된 각 속성이 나열됩니다. 예제:

이전 편집을 기반으로 시스템에서 원래 생성한 결과를 다음과 같이 변경했습니다. 위험 수준이 중간에서 중요로 변경됨, 위험 점수가 5.3에서 9.5로 변경됨, 근거: 공개적으로 액세스할 수 있는 소스 맵을 통해 애플리케이션 소스 코드의 전체 노출을 반영하도록 심각도를 업그레이드했습니다.

1. 실행 방법을 결정하기 전에 요약을 검토하여 조사 결과가 어떻게 조정되었는지 이해합니다.
2. 개인화된 조사 결과를 재정의하려면 다른 조사 결과와 마찬가지로 다시 편집합니다(예: 심각도 또는 위험 점수 변경). 최신 편집이 항상 우선합니다. AWS Security Agent는 이를 통해 학습하고 다음 실행 시 업데이트된 기본 설정을 적용하여 이전 규칙을 대체합니다.

> **Note**
>
> 개인화 변경은 편집에서 AWS Security Agent가 학습한 표준을 반영하도록 riskLevel, riskScore, 이름, 설명, 추론 및 attackScript와 같은 결과 속성을 조정합니다. 이전에 편집한 모든 필드는 향후 실행에서 유사한 결과에 따라 조정될 수 있습니다.

개인 맞춤이 편집에서 학습하는 방법

조사 결과 개인화가 활성화되면 AWS Security Agent는 조사 결과에 대한 모든 편집 내용을 학습하고 향후 실행에 유사한 조정을 적용합니다. 편집하는 모든 필드(이전 결과 편집 섹션에 나열됨)는 에이전트가 학습한 기본 설정을 구체화하는 데 사용됩니다.

> **Note**
>
> 상태 필드의 경우 조사 결과를 FALSE_POSITIVE로 표시하는 것만 학습 가능한 기본 설정으로 처리됩니다. 다른 상태 값을 변경해도 학습이 트리거되지 않습니다.

다음 펜테스트 실행 시 에이전트는 학습한 기본 설정을 기준으로 새 결과를 평가하고 해당하는 경우 조정을 적용합니다. 조정된 모든 결과에 개인화 변경 섹션이 나타나 변경 사항을 설명합니다.

> **Note**
>
> 개인화는 가장 최근에 완료된 Pentest 실행에서 편집한 내용에서만 학습합니다. 학습을 위해 pentest가 시작될 때 기능을 활성화해야 합니다. 이후 실행에서 동일한 결과를 편집하는 경우 가장 최근 편집이 우선합니다. 에이전트는 최신 결정을 반영하도록 기본 설정을 업데이트합니다.

결과 개인화 활성화 또는 비활성화

결과 개인화는 기본적으로 활성화되어 있습니다. 비활성화하거나 다시 활성화하려면:

1. pentest 구성으로 이동합니다.
2. 결과 개인화 토글을 찾습니다.
3. 결과 개인화를 활성화하려면 토글을 켭니다. 비활성화하려면 끕니다.

비활성화하면 조사 결과가 개인 맞춤이 적용되지 않고 원시 에이전트 생성 상태로 나타납니다. 이전에 학습한 기본 설정은 유지되며 기능이 다시 활성화된 경우 다시 적용됩니다.

7단계: CVSS 지표 해석

CVSS 지표를 이해하면 실제 심각도를 평가하고 문제 해결 작업의 우선순위를 정하는 데 도움이 됩니다.

추론 섹션을 검토할 때 다음 주요 지표에 유의하세요.

- 공격 벡터(Network/Adjacent/Local/Physical) - 공격을 원격으로 실행할 수 있는 방법을 나타냅니다.
- 공격 복잡성(낮음/높음) - 악용에 특수 조건이 필요한지 여부를 보여줍니다.
- 필요한 권한(None/Low/High음) - 공격자에게 필요한 액세스 수준을 식별합니다.
- 사용자 상호 작용(없음/필수) - 악용에 사용자 개입이 필요한지 여부를 결정합니다.
- Confidentiality/Integrity/Availability 영향(None/Low/High음) - 시스템 보안에 미치는 영향을 측정합니다.

> **Important**
>
> 네트워크 공격 벡터, 낮은 복잡성 및 높은 기밀성/무결성 영향이 있는 결과는 즉각적인 주의가 필요한 가장 위험한 취약성을 나타냅니다.

8단계: 조사 결과 우선순위 지정 및 해결

조사 결과에 대한 조치를 취하여 취약성을 해결하고 애플리케이션의 보안 태세를 개선합니다.

신뢰도가 높은 중요 및 높은 심각도 조사 결과의 경우:

1. 설명 및 단계를 검토하여 섹션을 철저히 재현합니다.
2. "작업 및 로그 추적" 링크를 통해 세부 로그에 액세스하여 전체 증거를 수집합니다.
3. 다음 방법 중 하나를 통해 ready-to-implement 가능한 코드 수정에 액세스합니다.
   - 자동 문제 해결의 경우: 문제 해결 섹션의 풀 요청 링크 사용
   - 수동 요청의 경우: 결과 페이지에서 '해결 코드'를 선택하여 풀 요청 사전 조건을 요청합니다.
   - 관리자는 AWS Security Agent 콘솔에서 GitHub 리포지토리에 대한 코드 수정을 활성화해야 합니다.

- 리포지토리는 Pentest 구성에 포함되어야 합니다.

4. 후속 침투 테스트를 계획하여 수정 사항이 효과적인지 확인합니다.

중간 및 낮은 심각도 조사 결과의 경우:

1. 위험 허용 범위 및 비즈니스 컨텍스트에 따라 우선순위를 정합니다.
2. 정기적인 개발 스프린트 계획에 문제 해결 작업을 포함합니다.
3. 심각도가 낮은 여러 조사 결과가 함께 더 높은 위험을 초래하는지 고려합니다.
4. 적절한 근거와 함께 수락된 위험을 문서화합니다.

> **Important**
>
> 심각도가 낮은 조사 결과는 무시하지 마세요. 특히 소셜 엔지니어링 또는 물리적 액세스와 결합된 경우 심각도가 낮은 여러 취약성을 함께 연결하여 더 심각한 악용을 생성할 수 있습니다.

9단계: 문제 해결 진행 상황 추적

조사 결과 인터페이스를 사용하여 해결된 취약성과 추가 조치가 필요한 취약성을 추적합니다.

1. 문제 해결 작업을 할 때 다시 재현 단계 섹션을 참조하여 수정 사항을 확인합니다.
2. 향후 참조 및 규정 준수 감사를 위해 각 결과에 대한 수정 접근 방식을 문서화합니다.

> **Tip**
>
> 취약성을 해결하기 위해 내린 코드 변경, 구성 업데이트 또는 아키텍처 결정을 포함하여 각 결과를 해결 방법에 매핑하는 문제 해결 로그를 유지 관리합니다.

다음 단계

침투 테스트 결과를 검토한 후:

- 즉각적인 해결을 위해 높은 신뢰도로 중요하고 심각도가 높은 조사 결과의 우선 순위 지정
- 테스트 환경에서 확인 스크립트를 다운로드, 검토 및 실행하여 스크립트를 테스트하고 취약성을 식별합니다.

- 결과 세부 정보 및 증거 링크를 사용하여 문제 관리 시스템에서 추적 티켓 생성
- 식별된 취약성을 해결하기 위한 수정 및 보안 제어 구현
- 새로 발견된 취약성에 대한 침투 테스트 실행 진행률 지표 모니터링
- 후속 침투 테스트를 예약하여 취약성이 제대로 해결되었는지 확인합니다.
- 조사 결과를 기반으로 애플리케이션 보안 테스트 프로세스 및 위협 모델 업데이트
- CVSS 지표를 검토하여 애플리케이션의 전반적인 보안 태세 이해

침투 테스트 수행에 대한 자세한 내용은 단원을 참조하십시오the section called “침투 테스트 생성”.

Security Agent 수명 주기 이해에 대한 자세한 내용은 섹션을 참조하세요the section called “리소스 계층 구조 및 수명 주기 이해”.

침투 테스트를 위한 인증 자격 증명 제공

AWS Security Agent가 웹 애플리케이션의 인증된 영역을 테스트할 수 있도록 자격 증명을 제공합니다. 자격 증명이 없으면 에이전트는 공개적으로 액세스할 수 있는 페이지 및 APIs.

인증 자격 증명 구성

1. 침투 테스트 생성 워크플로에서 인증 자격 증명 - 선택 사항 섹션을 찾습니다.
2. 자격 증명 #1 섹션에서 자격 증명 입력 방법을 선택합니다.
   - 자격 증명 입력 - 자격 증명을 직접 입력합니다. 개발 및 테스트 환경에 가장 적합합니다.
   - 고급 설정 - AWS 네이티브 자격 증명 관리를 사용합니다. 프로덕션 환경 및 민감한 자격 증명에 권장됩니다.

고급 옵션

고급 설정을 선택하면 세 가지 자격 증명 전략 중에서 선택할 수 있습니다.

- IAM 역할 가정 - AWS Cognito 또는 IAM 인증을 사용하는 애플리케이션의 경우
- AWS Secrets Manager - 암호화 및 교체를 통한 보안 자격 증명 스토리지용
- Lambda 함수 - 동적 자격 증명 생성 또는 복잡한 인증 흐름용

자격 증명을 직접 입력

1. 입력 자격 증명을 선택합니다.

2. 사용자 이름과 암호를 입력합니다.
3. 액세스 URL 드롭다운에서 이러한 자격 증명을 사용할 URL을 선택합니다. 대상 엔드포인트 목록에서 선택해야 합니다.
4. (선택 사항) 2FA - 선택 사항 필드에서 2단계 인증이 필요한 애플리케이션에 대한 TOTP 암호를 제공합니다. 다음 작업 중 하나를 수행할 수 있습니다.
   - TOTP 보안 암호를 직접 입력하거나(예: `JBSWY3DPEHPK3PXP`) 전체 `otpauth://totp/` URI를 입력합니다(예: `otpauth://totp/Example:user@example.com?secret=JBSWY3DPEHPK3PXP&issuer=Example`).
   - 업로드 아이콘을 선택하여 인증 앱 설정 페이지에서 QR 코드 이미지를 업로드합니다. QR 코드는 로컬에서 스캔되고 TOTP URI는 자동으로 추출됩니다.

     TOTP 보안 암호가 제공되면 에이전트는 새 일회용 코드를 자동으로 생성하고 로그인 중에 2FA 프롬프트가 감지되면 입력합니다.
5. (선택 사항) 애플리케이션에 복잡한 인증 흐름이 있는 경우 에이전트 스페이스 로그인 프롬프트를 확장하여 특정 로그인 지침을 제공합니다.

> **Important**
>
> 개인 또는 관리 계정이 아닌 대표 액세스 권한이 있는 테스트 계정을 사용합니다.

고급 설정 사용

고급 자격 증명 전략(Secrets Manager, Lambda 또는 IAM 역할)을 선택하면 AWS Security Agent는 서비스 역할을 사용하여 구성된 AWS 리소스에서 직접 자격 증명을 검색합니다. 에이전트 스페이스 로그인 프롬프트를 사용하여 에이전트에게 애플리케이션에 자격 증명을 적용하는 방법에 대한 지침을 제공합니다. 예를 들어 탐색할 로그인 URL과 채울 양식 필드 등이 있습니다.

> **Note**
>
> Secrets Manager 보안 암호 및 Lambda 함수는 AWS Security Agent 설정과 동일한 AWS 계정에 있어야 합니다. 교차 계정 자격 증명은 현재 지원되지 않습니다.

1. 고급 설정을 선택합니다.
2. 사용자 액세스 전략 드롭다운에서 다음 중 하나를 선택합니다.

에이전트가 수임할 사용 가능한 IAM 역할 선택

AWS Cognito, IAM 인증이 포함된 API Gateway 또는 기타 AWS 네이티브 인증 시스템을 사용하는 애플리케이션에이 옵션을 사용합니다. IAM 역할에는 AWS Security Agent가 이를 수임할 수 있는 신뢰 관계와 애플리케이션의 인증 시스템에 액세스할 수 있는 권한이 있어야 합니다.

연결된 AWS Secrets Manager에서 정적 자격 증명 선택

암호화, 교체 및 액세스 감사를 통해 AWS Secrets Manager에서 보안 인증을 안전하게 검색하려면이 옵션을 사용합니다.

IAM 역할에는 `secretsmanager:GetSecretValue` 및 `secretsmanager:DescribeSecret` 권한이 있어야 합니다.

에이전트는 Secrets Manager에서 직접 보안 암호 값을 검색합니다. 에이전트 스페이스 로그인 프롬프트를 사용하여 에이전트에게 애플리케이션의 로그인 흐름에 자격 증명을 적용하는 방법을 알려줍니다. 예를 들어 탐색할 URL과 채울 양식 필드 등이 있습니다. 에이전트가 로그인 프롬프트에 제공한 지침에 따라 형식을 해석하므로 모든 형식을 사용하여 보안 암호를 저장할 수 있습니다.

예를 들어 에이전트가 https://example.com/login 사용자 이름/암호 로그인 양식을 제출하는 경우 `username` 및 `password` 필드를 사용하여 보안 암호를 JSON으로 포맷할 수 있습니다. 애플리케이션에 TOTP 기반 2FA가 필요한 경우 TOTP 보안 암호가 직접 있거나 전체 `otpauth://totp/` URI가 있는 `totpSecret` 필드를 포함합니다.

```
{
  "username": "test-user",
  "password": "secure-password-here",
  "totpSecret": "JBSWY3DPEHPK3PXP"
}
```

그런 다음 인증 지침을 구성합니다. 액세스 URL을 `https://example.com` (또는 대상 엔드포인트 목록에서 선택한 다른 URL)로 설정합니다. 에이전트 스페이스 로그인 프롬프트에 "https://example.com/login 이동하여 제공된 사용자 이름과 암호를 양식에 입력합니다."를 입력합니다.

또 다른 예로, HTTP 헤더에 제공할 API 키가 있는 경우 일반 텍스트로 저장할 수 있습니다.

```
"api-key-here"
```

그런 다음 인증 지침을 구성합니다. 에이전트 스페이스 로그인 프롬프트에 "모든 요청에 대해 X-API-Key 헤더를 제공된 API 키로 설정"을 입력합니다.

> **Important**
>
> TOTP 기반 2FA만 지원됩니다. SMS, 이메일, 푸시 알림, 하드웨어 키 및 OAuth 인증은 지원되지 않습니다.

사용 가능한 Lambda 함수를 선택하여 자격 증명을 동적으로 검색

복잡한 인증 시스템, 동적 자격 증명 생성 또는 외부 자격 증명 공급자와의 통합에이 옵션을 사용합니다.

IAM 역할에는 `lambda:InvokeFunction` 권한이 있어야 하며 함수는 30초 이내에 완료되어야 합니다.

침투 테스트가 실행되면 AWS Security Agent는 AWS Lambda 함수를 동기적으로 호출하고 침투 테스트와 해결 중인 특정 자격 증명을 식별하는 이벤트를 전달합니다.

```
{
  "pentest_arn": "arn:aws:securityagent:us-west-2:111122223333:pentest/pt-
abcd1234-5678-90ab-cdef-EXAMPLE11111",
  "actor_identifier": "Credential1"
}
```

- `pentest_arn` - 자격 증명이 확인되는 침투 테스트의 Amazon 리소스 이름(ARN)입니다. 함수 내에서 요청의 범위를 지정, 권한 부여 또는 감사하는 데 사용합니다.
- `actor_identifier` - 콘솔에서이 자격 증명에 할당한 자격 증명 이름입니다(예: `Credential1`). 단일 함수가 여러 자격 증명을 제공할 때 올바른 자격 증명을 반환하는 데 사용합니다.

에이전트는 함수의 출력을 자격 증명으로 직접 사용합니다. 에이전트 스페이스 로그인 프롬프트를 사용하여 AWS Secrets Manager와 동일한 방식으로 애플리케이션에 이러한 자격 증명을 적용하는 방법을 에이전트에게 알립니다. Lambda 함수 및 지원되는 인증 유형의 출력을 포맷하는 방법에 the section called “연결된 AWS Secrets Manager에서 정적 자격 증명 선택” 대한 예는 섹션을 참조하세요.

여러 자격 증명 구성

다양한 사용자 역할 또는 인증 시스템을 테스트하려면:

1. 다른 자격 증명 추가를 선택합니다.
2. 두 입력 방법 중 하나를 사용하여 추가 자격 증명을 구성합니다.

3. 자격 증명을 제거하려면 자격 증명 섹션에서 제거를 선택합니다.

로그인 최적화

로그인 최적화를 통해 AWS Security Agent는 이전 Pentest 실행의 탐색 패턴을 학습하고 후속 실행에 적용할 수 있습니다. 이러한 패턴에는 로그인 흐름과 다단계 인증 워크플로가 포함됩니다. 이렇게 하면 에이전트가 인증 방법을 다시 검색하는 데 소요되는 시간이 줄어들어 스캔 속도가 빨라지고 테스트 적용 범위가 더 일관되게 유지됩니다.

로그인 최적화 작동 방식

첫 번째 Pentest 실행에서 에이전트는 사용자가 제공한 자격 증명을 사용하여 애플리케이션의 로그인 흐름을 탐색하는 방법을 발견합니다. 성공적인 탐색 단계를 기록하고 학습된 로그인 워크플로를 캡처하는 스킬 파일을 생성합니다.

후속 실행 시 에이전트는 저장된 스킬 파일을 읽고 학습된 탐색 패턴을 직접 적용하여 검색 단계를 건너뜁니다. 이는 다음을 의미합니다.

- 인증 테스트 시간 단축 - 에이전트는 처음부터 탐색하는 대신 검증된 탐색 패턴을 즉시 적용합니다.
- 보다 일관된 동작 - 에이전트는 대안을 탐색하는 대신 동일한 검증된 경로를 따릅니다.

> **Note**
>
> 로그인 최적화는 실행 내 첫 번째 로그인 세션 중에 학습합니다. 동일한 실행의 후속 로그인 세션은 첫 번째 세션에서 학습한 내용을 활용할 수 있습니다. 향후 실행 시 모든 로그인 세션은 저장된 스킬의 이점을 누릴 수 있습니다.

로그인 최적화 활성화 또는 비활성화

로그인 최적화는 기본적으로 활성화됩니다. 비활성화하거나 다시 활성화하려면:

1. pentest 구성에서 인증 자격 증명 - 선택 사항 섹션으로 이동합니다.
2. 로그인 최적화 토글을 찾습니다.
3. 로그인 최적화를 활성화하려면 토글을 켭니다. 비활성화하려면 끕니다.

비활성화하면 에이전트가 실행할 때마다 로그인 흐름을 처음부터 다시 검색합니다. 이전에 학습한 탐색 기술은 보존되며 기능이 다시 활성화된 경우 다시 적용됩니다.

> **Tip**
>
> 애플리케이션의 로그인 흐름이 크게 변경되면(예: 재설계된 로그인 페이지 또는 새 인증 단계) 에이전트는 다음 실행 시 학습된 스킬을 업데이트하여 자동으로 적응합니다. 최적화를 수동으로 재설정할 필요가 없습니다.

침투 테스트 결과 수정

침투 테스트 결과를 볼 때 AWS Security Agent에 결과 수정 시도를 요청할 수 있습니다. 프라이빗 GitHub 리포지토리의 경우 AWS Security Agent는 제안된 수정 사항이 포함된 풀 요청을 엽니다. 퍼블릭 리포지토리의 경우 로컬에서 적용할 수 있는 다운로드 가능한 diff 파일로 문제 해결을 사용할 수 있습니다.

AWS Management Console에서 결과 수정을 활성화해야 합니다. (the section called “사용자가 침투 테스트 및 코드 검토 결과의 수정을 시작할 수 있도록 지원”를 참조하세요.) 사용자는 AWS Security Agent 웹 앱에서 특정 결과에 대한 문제 해결을 시작할 수 있습니다.

사전 조건

시작하기 전에 다음을 갖추었는지 확인하세요.

- 완료되었거나 진행 중인 침투 테스트 실행
- AWS Security Agent 웹 애플리케이션에 대한 액세스
- 애플리케이션의 아키텍처 및 보안 요구 사항에 대한 지식

1단계: 자동 문제 해결 활성화 또는 비활성화

침투 테스트를 생성하거나 수정할 때 코드 수정 옵션을 구성할 수 있습니다. 자동 수정을 활성화하면 에이전트가 검사 중에 결과를 확인하면 AWS Security Agent는 연결된 GitHub 리포지토리를 자동으로 수정하려고 시도합니다. 코드 수정을 수동으로 시작할 수도 있습니다. 침투 테스트 세부 정보를 편집하는 보기의 자동 코드 수정 섹션에서 코드 수정을 활성화하거나 비활성화합니다.

2단계: 코드 수정을 위한 리포지토리 선택

1. 마지막 단계 추가 학습 리소스까지 다음을 선택합니다.
2. 리소스에서 선택을 선택합니다.
3. GitHub 리포지토리를 선택합니다.

4. 코드 수정에 사용할 리포지토리를 선택합니다.
5. 침투 테스트를 저장합니다.
6. 침투 테스트 학습 리소스 탭에서 성공적으로 연결된 리포지토리를 볼 수 있습니다.

3단계: 침투 테스트 시작 및 조사 결과 보기

침투 테스트를 실행하여 결과를 감지합니다. 자세한 내용은 the section called “침투 테스트 결과 검토” 단원을 참조하십시오.

4단계: 코드 수정 시작 및 보기

1. 조사 결과로 이동합니다.
2. 자동 코드 수정을 활성화한 경우 AWS Security Agent가 결과를 확인하면 코드 수정이 시작됩니다.
3. 코드 수정을 수동으로 시작하려면 코드 수정 버튼을 선택합니다.
4. 결과의 코드 수정 섹션에서 코드 수정 상태와 풀 요청에 대한 링크를 볼 수 있습니다. GitHub 리포지토리가 퍼블릭인 경우 풀 요청 대신 코드 수정을 다운로드 가능한 파일로 사용할 수 있습니다. 를 실행`git apply /path/to/code_remediation_changes.diff`하여 로컬에서 리포지토리에 변경 사항을 적용할 수 있습니다.

보안 및 참조

AWS Security Agent에 대한 보안 지침, 서비스 할당량 및 문서 기록.

AWS Security Agent의 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 이를 클라우드의 보안과 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS Security Agent를 실행하는 인프라를 보호할 책임이 있습니다. 여기에는 서비스 인프라, AI 모델 및 침투 테스트 에이전트가 포함됩니다. 서드 파티 감사원은 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다.
- 클라우드 내부의 보안 - 고객의 책임에는 다음 영역이 포함됩니다.
  - IAM 정책 및 권한을 통해 AWS Security Agent에 대한 액세스 관리
  - 설계 문서, 코드 리포지토리, 침투 테스트를 위한 애플리케이션 URLs 등 서비스에 제공하는 콘텐츠 보호
  - 모니터링할 리포지토리 및 애플리케이션 구성
  - 서비스에서 제공하는 보안 조사 결과 검토 및 조치
  - 제공된 문제 해결 지침에 따라 애플리케이션 보호
  - 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정

이 설명서는 AWS Security Agent를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다.

AWS Security Agent 및 AI 지원 침투 테스트를 위한 보안 고려 사항

AWS Security Agent는 모든 환경의 개발 수명 주기 동안 애플리케이션을 사전에 보호하는 프론티어 에이전트입니다. 보안 팀은 검토 중에 자동으로 검증되는 표준을 중앙에서 정의하여 요구 사항에 맞는 자동화된 보안 검토를 수행합니다. Security Agent는 애플리케이션에 맞게 사용자 지정된 온디맨드 침투 테스트를 수행하여 확인된 보안 위험을 검색하고 보고합니다. 이 접근 방식은 애플리케이션 전반의 보안 전문 지식을 확장하여 개발 속도와 일치시키는 동시에 포괄적인 보안 범위를 제공합니다. 설계에서 배포에 이르기까지 보안을 통합하면 취약성을 조기에 대규모로 방지하는 데 도움이 됩니다.

보안 팀은 AWS 콘솔에서 승인된 권한 부여 라이브러리, 로깅 표준 및 데이터 액세스 정책 등 조직 보안 요구 사항을 한 번 정의합니다. AWS Security Agent는 개발 전반에 걸쳐 이러한 보안 요구 사항을 자동으로 적용하여 아키텍처 문서 및 코드를 표준에 따라 평가하고 위반을 감지할 때 특정 지침을 제공합니다. 이렇게 하면 팀 간에 일관된 보안이 적용되고 개발 속도에 맞게 검토 규모가 조정됩니다.

배포 검증을 위해 AWS Security Agent는 침투 테스트를 주기적 병목 현상에서 온디맨드 기능으로 변환합니다. 보안 팀은 대상 URLs, 인증 세부 정보, 소스 코드 및 설명서를 제공합니다. AWS Security Agent는 심층적인 애플리케이션 이해를 개발하고 정교한 공격 체인을 실행하여 취약성을 발견하고 검증하므로 팀이 필요할 때마다 테스트할 수 있습니다.

주요 기능

AWS Security Agent는 전체 개발 수명 주기에 걸쳐 포괄적인 보안 기능을 제공합니다.

보안 검토 설계

AWS Security Agent는 설계 문서에 대한 온디맨드 보안 피드백을 제공하고 코드가 작성되기 전에 조직 보안 요구 사항 준수를 평가합니다. 보안 팀은 웹 애플리케이션을 통해 설계 문서를 업로드합니다. 여기서 에이전트는 보안 요구 사항을 기준으로 이를 분석하고 문제 해결 지침에 따라 조사 결과를 표시합니다. 이렇게 하면 몇 시간 동안의 수동 검토를 집중 분석으로 가속화하여 문제 해결이 가장 효율적일 때 팀이 보안 문제를 해결할 수 있습니다.

코드 보안 검토

AWS Security Agent는 풀 요청 또는 업로드된 코드를 분석하여 조직 보안 요구 사항과 입력 검증 누락 및 SQL 주입 위험과 같은 일반적인 보안 문제를 확인합니다. 에이전트는 코드 리포지토리 플랫폼 내에서 직접 문제 해결 지침을 제공합니다. 보안 팀은 모니터링할 리포지토리를 구성하여 중요한 문제에 대한 감독을 유지하면서 모든 코드베이스에서 평가를 확장합니다.

온디맨드 침투 테스트

AWS Security Agent는 맞춤형 다단계 공격 시나리오를 통해 검증된 보안 취약성을 검색하고 보고하는 온디맨드 침투 테스트를 제공합니다. AWS Security Agent는 제공된 설명서 및 자격 증명에서 애플리케이션 컨텍스트를 개발한 다음 정교한 공격 체인을 실행하여 기존 도구에서 놓치는 복잡한 취약성을 식별하는 특수 AI 에이전트를 배포합니다. 영향 분석, 재현 가능한 공격 경로 및 ready-to-implement 가능한 코드 수정을 통해 결과를 문서화하여 침투 테스트를 몇 주에서 몇 시간으로 가속화하고 애플리케이션 포트폴리오 전반에서 검증을 확장합니다.

FAQ

보안 및 제어

AWS Security Agent는 시스템에 대한 액세스를 어떻게 인증하고 유지하나요?

침투 테스트는 런타임 시 사용자의 시스템에 인증할 수 있는 AWS Security Agent의 유일한 기능입니다. AWS Security Agent는 펜 테스트를 시작하기 전에 정적 사용자 이름 및 암호 자격 증명(Secrets Manager에 저장됨) 또는 자격 증명 공급업체(Lambda 함수) 형식의 자격 증명을 구성으로 수락합니다. 이러한 자격 증명은 펜 테스트의 수명 주기 동안 사용자 시스템/애플리케이션의 정상적인 기능을 연습하는 데 사용됩니다. Pentest를 위해 적절한 범위의 권한으로 새 자격 증명을 생성하는 것이 좋습니다.

사용자가 테스트 범위와 깊이를 제어하여 의도하지 않은 시스템 영향을 방지할 수 있나요?

AWS Security Agent를 사용하면 고객이 엔드포인트에서 탐색할 특정 취약성 범주를 선택할 수 있습니다. 사용자는 out-of-scope URLs 지정하여 AWS Security Agent가 해당 대상에 대해 침투 테스트를 수행하지 못하도록 할 수 있습니다. https://docs.aws.amazon.com/securityagent/latest/userguide/perform-penetration-test.html

AWS Security Agent 자체가 보안 위험을 초래할 수 있나요?

AWS Security Agent는 보안 위험을 발견하지만 의도적으로 영향을 미치는 페이로드를 최소화(예: SQL 주입 공격이 발견될 때 테이블을 삭제하는 대신 SQL 버전 추출)하여 이를 수행하도록 지시합니다. 또한 AWS Security Agent는 대상 애플리케이션에 대한 과도한 부하 생성과 같은 위험한 동작을 방지하기 위해 결정적 가드레일로 제한됩니다. 가드레일이 마련되어 있지만 의도하지 않거나 명확하지 않은 비즈니스 로직 상호 작용이 있을 수 있으므로 항상 사전 프로덕션 환경에 대해 침투 테스트를 수행하는 것이 좋습니다.

AWS Security Agent는 어떤 데이터를 수집하며 어디에 저장되나요?

AWS Security Agent를 사용하면 사용자가 아티팩트를 업로드하여 테스트 중인 애플리케이션에 대한 컨텍스트를 제공할 수 있습니다. 데이터 보호에 대한 자세한 내용은 섹션을 참조하세요the section called “데이터 보호”. AWS Security Agent는 추론 요청을 처리하기 위해 리전 내에서 최적의 리전을 자동으로 선택합니다. 이렇게 하면 사용 가능한 컴퓨팅 리소스와 모델 가용성이 극대화되고 최상의 고객 경험이 제공됩니다. 데이터는 요청이 시작된 리전에만 저장되지만 입력 프롬프트 및 출력 결과는 해당 리전 외부에서 처리될 수 있습니다. 모든 데이터는 Amazon의 보안 네트워크를 통해 암호화되어 전송됩니다. 자세한 내용은 교차 리전 추론을 참조하세요.

엔드포인트에 대한 무단 테스트를 차단하기 위해 어떤 제어가 있습니까?

Pentesting을 위한 대상 URLs로 지정된 엔드포인트는 소유권의 척도로 DNS 검증 또는 HTTP 검증이 필요합니다. AWS Security Agent는 고객에게 엔드포인트의 DNS에 TXT 레코드를 추가하거나 HTTP Route 반환 검증 문자열을 소유권 증명으로 노출하도록 요청합니다. 소유권 증명을 보여준 후에만 사용자는 Pentest를 진행할 수 있습니다. 대상 외부의 URLs 및 액세스 가능한 URLs에 대한 요청은 네트워크에 의해 차단됩니다.

고객은 침투 테스트 활동의 영향을 받을 수 있는 모든 시스템을 테스트할 수 있는 적절한 권한이 있는지 확인할 책임이 있습니다. AWS Security Agent의 모든 사용은 AWS 허용 사용 정책(https://aws.amazon.com/aup/)을 준수해야 합니다.

사용자는 AWS Security Agent를 사용하여 침해를 어떻게 차단하고 보고하나요?

AWS Security Agent는 요청을 지속적으로 모니터링하고 대상 URLs 있는 URLs. AWS Security Agent를 사용하여 타사 엔드포인트에 대한 무단 테스트를 수행하는 등 침해가 감지되면 계정에서 진행 중인 모든 Pentest가 종료됩니다. 고객은 AWS Support 또는 AWS 계정 팀에 도움을 요청할 수 있습니다.

AWS Security Agent가 펜 테스트 워크플로를 교체할 수 있나요?

AWS Security Agent는 전문적인 침투 테스트 서비스가 아니므로 사용자가 AWS Security Agent를 보안 검토 워크플로에 통합하는 것이 좋습니다. AWS Security Agent는 Pentesting 전문가와 협력하는 것이 너무 이르거나 실용적이지 않거나 너무 자주 재평가해야 하는 경우 소프트웨어 수명 주기의 개발 단계에서 온디맨드 방식으로 침투 테스트에 대한 접근성을 제공할 수 있습니다. 보안 전문가는 AWS Security Agent의 조사 결과를 검토하여 검증하거나, 설명하거나, 새로운 조사 결과(있는 경우)를 얻을 수 있습니다.

사용자가 다른 팀원에 대해 역할 기반 액세스 제어(RBAC)를 설정할 수 있나요?

예. AWS Security Agent는 AWS IAM Identity Center와 통합되므로 관리자는 AWS Security Agent 웹 애플리케이션에 액세스할 수 있는 팀원을 관리할 수 있습니다. 이를 통해 사용자는 설계 검토 및 Pentest를 생성, 관리 및 볼 수 있습니다.

테스트 기능

AWS Security Agent가 탐지할 수 있는 취약성 유형은 무엇입니까?

AWS Security Agent는 웹 애플리케이션의 OWASP 상위 10개에서 취약성을 탐지합니다. AWS Security Agent는 아래에 설명된 테스트에 포함하거나 제외할 수 있는 특정 위험 유형을 제공합니다. 결과는 이러한 위험 범주 내에서 발생하거나 이러한 위험 범주의 조합에서 다음 리드가 발견한 새로운 결과에서 발생할 수 있습니다.

- 임의 파일 업로드
  - 임의 파일 업로드는 애플리케이션이 애플리케이션과 사용자를 안전하게 유지하는 방식으로 가짜 파일과 악성 파일을 방어할 수 있어야 함을 확인합니다.
- 코드 삽입
  - 코드 주입은 애플리케이션에서 해석/실행하는 코드 주입으로 구성된 공격 유형의 일반 용어입니다.
- 명령 주입
  - 명령 주입은 취약한 애플리케이션을 통해 호스트 운영 체제에서 임의 명령을 실행하는 것을 목표로 하는 공격입니다.
- 교차 사이트 스크립팅(XSS)
  - 교차 사이트 스크립팅(XSS) 공격은 악성 스크립트가 악의적이고 신뢰할 수 있는 웹 사이트에 삽입되는 주입의 한 유형입니다.
- 안전하지 않은 직접 객체 참조
  - 안전하지 않은 직접 객체 참조(IDOR)는 애플리케이션이 사용자 제공 입력을 기반으로 객체에 대한 직접 액세스를 제공할 때 발생합니다.
- JSON 웹 토큰 취약성
  - JWTs는 애플리케이션과 기본 라이브러리 모두에서 구현되는 일반적인 취약성 소스입니다.
- 로컬 파일 포함
  - 파일 포함 취약성을 통해 공격자는 파일을 포함할 수 있으며, 일반적으로 대상 애플리케이션에 구현된 “동적 파일 포함” 메커니즘을 악용합니다.
- 경로 순회
  - 경로 순회 공격(디렉터리 순회라고도 함)은 웹 루트 폴더 외부에 저장된 파일 및 디렉터리에 액세스하는 것을 목표로 합니다.
- 권한 에스컬레이션
  - 권한 에스컬레이션은 사용자가 일반적으로 허용되는 것보다 더 많은 리소스 또는 기능에 액세스할 수 있고 애플리케이션에서 이러한 상승 또는 변경을 방지해야 할 때 발생합니다.
- 서버 측 요청 위조(SSRF)
  - 서버 측 요청 위조(SSRF)는 공격자가 서버의 기능을 남용하여 내부 리소스를 읽거나 업데이트할 수 있을 때 발생합니다.
- 서버 측 템플릿 주입
  - 서버 측 템플릿 주입 취약성(SSTI)은 사용자 입력이 안전하지 않은 방식으로 템플릿에 포함되어 서버에서 원격 코드 실행이 발생할 때 발생합니다.

- SQL 주입
  - SQL 삽입 공격은 클라이언트에서 애플리케이션으로의 입력 데이터를 통한 SQL 쿼리의 삽입 또는 “삽입”으로 구성됩니다.
- XML 외부 엔터티
  - XML 외부 개체 공격은 XML 입력을 구문 분석하는 애플리케이션에 대한 공격의 한 유형입니다. 이 공격은 외부 개체에 대한 참조가 포함된 XML 입력이 약하게 구성된 XML 구문 분석기에 의해 처리될 때 발생합니다.

AWS Security Agent는 어떤 인증 방법을 지원하나요?

AWS Security Agent는 OAuth 및 JWT를 포함한 일반적인 인증 방법을 지원합니다. 자세한 내용은 설명서를 참조하세요.

AWS Security Agent는 속도 제한 및 서비스 거부(DOS) 방지를 어떻게 처리하나요?

AWS Security Agent에는 DOS를 포함하여 테스트 중인 엔드포인트를 중단하거나 중단하지 않도록 하는 가드레일이 있습니다. 여기에는 예상치 못한 트래픽 패턴을 감지하고 처리할 수 있는 내부 속도 제어 기능이 있습니다.

AWS Security Agent가 REST 및 GraphQL APIs 모두 테스트할 수 있나요?

예, AWS Security Agent는 API 엔드포인트를 테스트할 수 있습니다. AWS Security Agent가 테스트 중인 각 API의 모양과 기능에 대해 더 나은 컨텍스트를 가질 수 있도록 API 설명서를 추가 학습 리소스로 제공하는 것이 좋습니다.

사용자는 AWS Security Agent가 모든 중요한 애플리케이션 로직 및 엔드포인트를 다루었는지 어떻게 확인할 수 있습니까?

AWS Security Agent는 대상 애플리케이션(들)을 폭넓게 탐색하고 악용하기 전에 이를 정상적으로 실행하려고 시도합니다. 이를 통해 런타임 시 애플리케이션에 대한 작업 이해를 구축하고 중요한 애플리케이션 로직 및 엔드포인트를 검색할 수 있습니다. 확률적 특성을 고려할 때 AWS Security Agent는 대상 애플리케이션에 대한 모든 중요한 애플리케이션 및 엔드포인트를 검색하고 테스트할 것을 보장하지 않습니다. AWS Security Agent 웹 애플리케이션은 검색된 모든 엔드포인트와 침투 테스트 로그에서 수행된 작업에 대한 가시성을 제공합니다.

정확도 및 신뢰성

AWS Security Agent는 보고하기 전에 조사 결과를 어떻게 검증하나요?

AWS Security Agent는 결정적 검사기를 사용하여 보고된 결과를 검증하는 데 도움이 됩니다. 결정적 검사기를 사용할 수 없는 위험 유형에서 AWS Security Agent는 독립적으로 결과 단계를 재생하여 결과의 유효성에 대한 확신을 얻습니다. AWS Security Agent는 신뢰도가 높거나 중간인 조사 결과만 보고하고 기본적으로 확인되지 않은 조사 결과를 숨깁니다.

AWS Security Agent가 사용자 지정 애플리케이션 로직에 맞게 조정할 수 있나요?

AWS Security Agent는 선택적으로 소스 코드, 위협 모델, 설계 문서 및 API 설명서를 추가 학습 리소스로 받아 Pentest의 수명 주기에 사용되는 대상 애플리케이션에 대한 사용자 지정 컨텍스트를 얻습니다.

사용자가 실행 전에 AWS Security Agent 테스트 방법론을 검토할 수 있나요?

현재 AWS Security Agent의 작업 과정을 미리 볼 수 있는 방법은 없습니다. AWS Security Agent 계획은 대상 애플리케이션의 탐색에 따라 본질적으로 동적입니다. 고객은 침투 테스트 로그를 관찰하여 실시간으로 탐색하는 동안 AWS Security Agent를 모니터링할 수 있습니다. 로그에 유효하지 않거나 바람직하지 않은 궤적이 표시되면 고객은 지속적인 Pentest 실행을 중지할 수 있습니다.

통합 및 배포

AWS Security Agent는 보안 도구(SIEM, 취약성 관리) 또는 CI/CD 파이프라인과 통합되나요?

AWS Security Agent는 기존 보안 도구 또는 CI/CD 파이프라인과 통합되지 않습니다.

AWS Security Agent는 환경별 구성을 어떻게 처리하나요?

AWS Security Agent는 특정 IAM 역할, VPCs 내부, 고객 지정 애플리케이션 관련 자격 증명, 대상 애플리케이션에 대한 소스 코드 참조로 Github 소스 리포지토리를 사용하여 실행되도록 구성할 수 있습니다.

AWS Security Agent는 에어 갭 또는 격리된 환경에서 실행할 수 있습니까?

아웃바운드 인터넷에 액세스할 수 없는 VPCs에 연결하도록 AWS Security Agent를 구성할 수 있습니다.

여러 팀원이 동시에 테스트를 실행할 수 있나요?

AWS Security Agent는 테스트를 시작하는 사람과 관계없이 계정당 5개의 동시 Pentest 실행을 지원합니다. 고객은 최대 100개의 에이전트 스페이스와 1,000개의 Pentest 프로젝트를 생성할 수 있습니다.

운영 영향

테스트된 시스템에 미치는 성능 영향은 무엇입니까?

AWS Security Agent에는 테스트 중인 엔드포인트가 중단되거나 중단되는 것을 방지하는 가드레일이 있습니다. 여기에는 AWS Security Agent가 엔드포인트에 수행할 수 있는 호출 수에 대한 속도 제어가 포함됩니다. 테스트 중인 시스템 또는 엔드포인트는 펜 테스트 활동으로 인해 트래픽이 약간 증가하고 잠재적 모니터링 알림이 트리거될 것으로 예상해야 합니다. AWS Security Agent 또는 모든 펜 테스트 활동은 사전 프로덕션 환경에서만 실행하는 것이 좋습니다.

사용자가 AWS Security Agent를 예약하거나 제한할 수 있나요?

AWS Security Agent에는 퍼블릭 APIs 펜 테스트 실행을 예약할 수 있는 기능이 없습니다. 또한 AWS Security Agent는 펜 테스트 실행을 시작할 때 대상 엔드포인트에 대한 요청에 대한 동시성 제어를 제공하지 않습니다. AWS Security Agent가 대상 엔드포인트에 문제를 일으키는 경우 고객은 진행 중인 pentest(s)를 중지할 수 있습니다.

전체 보안 평가의 일반적인 기간은 어떻게 됩니까?

각 pentest의 런타임은 대상 애플리케이션의 범위와 평가하도록 구성된 위험 유형에 따라 달라집니다. 대부분의 pentest 실행은 16시간 이내에 완료됩니다.

AWS Security Agents에 대한 AWS 관리형 정책

AWS 관리형 정책은 AWS에서 생성 및 관리하는 독립적 정책입니다. 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. AWS가 AWS 관리형 정책에 정의된 권한을 업데이트하는 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 줍니다. AWS는 새 AWS 서비스가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: SecurityAgentWebAppAPIPolicy

Security Agent 웹 애플리케이션 API와 상호 작용할 수 있는 권한을 부여합니다. 이 정책을 통해 사용자는 자동화된 보안 침투 테스트를 구성 및 실행하고, 테스트 실행을 관리하고, 보안 결과를 보고,

Security Agent 리소스에 액세스할 수 있습니다. 이 정책은 레거시 에이전트 인스턴스 리소스 유형과 특정 레거시 IAM 작업을 참조합니다.

권한 세부 정보

이 정책은 다음에 대해 Security Testing Control 서비스(securityagent:*)와 상호 작용할 수 있는 권한을 부여합니다.

- Pentest Management: 침투 테스트 및 실행 작업 생성, 업데이트, 삭제 및 나열
- 보안 조사 결과: 관련 콘텐츠 및 메타데이터를 포함하여 완료된 테스트의 보안 조사 결과를 보고, 설명하고, 업데이트합니다.
- 작업 관리: 코드 검토 및 설명서 검토 작업 나열 및 검색
- 리소스 검색: 에이전트 인스턴스, 아티팩트, 통합 및 검색된 엔드포인트 나열 및 보기
- 테스트 실행: 실시간 모니터링 기능을 사용하여 Pentest 실행 시작 및 중지
- 코드 수정: 보안 조사 결과에 대한 자동 코드 수정 시작

최신 버전의 JSON 정책 문서를 보려면 AWS 관리형 정책 참조 안내서의 SecurityAgentWebAppAPIPolicy를 참조하세요.

AWS 관리형 정책: AWSSecurityAgentWebAppPolicy

Security Agent 웹 애플리케이션 API와 상호 작용할 수 있는 권한을 부여합니다. 이 정책을 통해 사용자는 자동화된 보안 침투 테스트를 구성 및 실행하고, 테스트 실행을 관리하고, 보안 결과를 보고, Security Agent 리소스에 액세스할 수 있습니다.

권한 세부 정보

이 정책은 다음에 대해 Security Testing Control 서비스(securityagent:*)와 상호 작용할 수 있는 권한을 부여합니다.

- Pentest Management: 침투 테스트 및 해당 작업 생성, 업데이트, 삭제 및 나열
- 보안 조사 결과: 관련 콘텐츠 및 메타데이터를 포함하여 완료된 테스트의 보안 조사 결과를 보고, 설명하고, 업데이트합니다.
- 작업 관리: 코드 검토 및 설계 검토 작업 나열 및 검색
- 리소스 검색: 에이전트 공간, 아티팩트, 통합 및 검색된 엔드포인트 나열 및 보기
- 테스트 실행: 실시간 모니터링 기능을 사용하여 pentest 작업 시작 및 중지

- 코드 수정: 보안 조사 결과에 대한 자동 코드 수정 시작

최신 버전의 JSON 정책 문서를 보려면 AWS 관리형 정책 참조 안내서의 AWSSecurityAgentWebAppPolicy를 참조하세요.

AWS Security Agents의 AWS 관리형 정책 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 AWS Security Agent의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다.

이 특정 설명서 페이지의 모든 소스 파일 변경 사항에 대한 알림을 받으려면 RSS 리더를 사용하여 다음 URL을 구독하세요.

변경	설명	Date
AWSSecurityAgentWebAppPolicy에 권한을 추가했습니다.	새 `DesignReviewFeedback` 리소스 유형에 대한 `TargetDomain` 및 리소스 권한이 추가되었습니다.	2026년 3월 31일
새 관리형 정책 AWSSecurityAgentWebAppPolicy가 추가되었습니다.	새 AgentSpace 리소스 유형 및 IAM 작업 이름 변경에 `AWSSecurityAgentWebAppPolicy` 대한 관리형 정책이 추가되었습니다.	2026년 2월 9일
SecurityAgentWebApAPIPolicy에 권한을 추가했습니다.	사용자가 보안 조사 결과에 대한 자동 코드 수정을 시작할 수 `securityagent:StartCodeRemediation` 있도록가 추가되었습니다.	2026년 1월 20일
SecurityAgentWebApAPIPolicy에 권한을 추가했습니다.	사용자가 콘솔에서 이미지를 볼 수 `securityagent:BatchGetSecurityTestContentMetadata` 있도록가 추가되었습니다.	2025년 12월 5일

변경	설명	Date
AWS Security Agents가 변경 사항 추적을 시작했습니다.	AWS Security Agents가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2025년 12월 2일

AWS Security Agent의 데이터 보호

AWS 공동 책임 모델은 AWS Security Agent의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호해야 합니다. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 또한 사용하는 AWS 서비스의 보안 구성 및 관리 작업에 대한 책임이 있습니다. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요. 데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 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.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요. CloudTrail
- AWS 암호화 솔루션을 AWS 제품 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 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 AWS Security Agent 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 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.

저장된 데이터 암호화

AWS Security Agent는 기본적으로 AWS 관리형 암호화 키를 사용하여 저장된 모든 데이터를 암호화합니다. 여기에는 다음이 포함됩니다.

- 설계 문서 및 코드 - 보안 검토를 위해 제공하는 모든 설계 문서, 코드 리포지토리 및 애플리케이션 아티팩트는 AES-256 암호화를 사용하여 암호화됩니다.
- 보안 조사 결과 - 모든 보안 조사 결과, 취약성 보고서 및 문제 해결 권장 사항은 유휴 시 암호화됩니다.
- 구성 데이터 - 보안 요구 사항, 사용자 지정 정책 및 서비스 구성이 암호화됩니다.
- 감사 로그 - 모든 서비스 활동 로그 및 감사 추적이 암호화됩니다.

AWS Security Agent는 AWS Key Management Service(AWS KMS)를 사용하여 암호화 키를 관리합니다. 선택적으로 고객 관리형 키를 사용하여 데이터를 암호화하여 리소스를 보호하는 암호화 키를 완벽하게 제어할 수 있습니다. 자세한 내용은 the section called “고객 관리형 키” 단원을 참조하십시오.

전송 중 데이터 암호화

AWS Security Agent는 전송 계층 보안(TLS) 1.2 이상을 사용하여 전송 중인 모든 데이터를 암호화합니다. 이는 다음에 적용됩니다.

- API 통신 - 애플리케이션과 AWS Security Agent 간의 모든 API 호출은 TLS 암호화와 함께 HTTPS를 사용합니다.
- 콘솔 액세스 - HTTPS를 통해 AWS Security Agent 콘솔에 액세스합니다.
- 리포지토리 연결 - GitHub 및 기타 코드 리포지토리에 대한 연결은 암호화된 프로토콜을 사용합니다.
- 에이전트 통신 - AWS Security Agent 서비스와 침투 테스트 에이전트 간의 모든 통신은 암호화된 채널을 사용합니다.

키 관리

AWS Security Agent는 AWS Key Management Service(AWS KMS)를 사용하여 암호화 키를 관리합니다. 기본적으로 데이터는 AWS 관리형 키를 사용하여 암호화됩니다. 선택적으로 에이전트 스페이스 및 통합과 같은 리소스를 생성할 때 고객 관리형 키를 지정할 수 있습니다. 자세한 내용은 the section called “고객 관리형 키” 단원을 참조하십시오.

인터네트워크 트래픽 개인 정보

AWS Security Agent는 퍼블릭 인터넷을 사용하여 클라우드 호스팅 소스 제어 공급자(GitHub, GitLab, Bitbucket) 및 Confluence Cloud와 통신합니다.

자체 호스팅 공급자(GitLab 자체 관리형, GitHub Enterprise Server)의 경우 모든 트래픽을 AWS 네트워크 내에 유지하도록 Amazon VPC Lattice를 사용하여 프라이빗 연결을 구성할 수 있습니다. 자세한 내용은 the section called “프라이빗 호스팅 소스 제어에 연결” 단원을 참조하십시오.

기본 구성에서 AWS Security Agent는 퍼블릭 인터넷을 사용하여 침투 테스트를 위해 앱에 연결합니다. VPC를 사용하여 애플리케이션에 액세스하도록 침투 테스트를 구성할 수도 있습니다. 자세한 내용은 the section called “에이전트를 프라이빗 VPC 리소스에 연결” 단원을 참조하십시오.

리전 간 데이터 처리

AWS Security Agent는 리전 간 추론을 사용하여 사용 가능한 컴퓨팅 리소스와 모델 가용성을 최적화합니다. 요청이 시작된 리전에 따라 입력 프롬프트를 처리하고 다른 리전에서 결과를 출력할 수 있습니다.

- 미국 동부(버지니아 북부) – us-east-1, 미국 서부(오레곤) – us-west-2, 아시아 태평양(시드니) – ap-southeast-2, 아시아 태평양(도쿄) – ap-northeast-1, 유럽(프랑크푸르트) – eu-central-1, 유럽(아일랜드) –에서 eu-west-1AWS Security Agent는 지리적 리전 간 추론을 사용합니다. 대부분의 기능에서 데이터 처리는 요청이 시작된 지리적 경계(예: 미국, EU, 호주 또는 일본) 내에 유지됩니다. 코드 수정의 경우 호주와 일본의 요청은 유럽 연합에서 처리됩니다. 기능별 라우팅 세부 정보는 교차 리전 추론 표를 참조하세요.
- 아시아 태평양(뭄바이) - ap-south-1, 아시아 태평양(싱가포르) - ap-southeast-1, 남아메리카(상파울루) -에서 sa-east-1AWS Security Agent는 글로벌 리전 간 추론을 사용합니다. 모든 상용 AWS 리전에서 입력 프롬프트 및 출력 결과를 처리할 수 있습니다.

어떤 경우에도 데이터는 요청이 시작된 리전에만 저장됩니다. 리전 간 작업 중에 전송되는 모든 데이터는 AWS 네트워크에 남아 있으며 퍼블릭 인터넷을 통과하지 않습니다. AWS 리전 간에 전송 중인 데이터를 암호화합니다.

교차 리전 추론은 항상 활성화되어 있으며 옵트아웃할 수 없습니다. 각 기능에 대한 요청을 처리하는 리전에 대한 자세한 내용은의 교차 리전 추론 섹션을 참조하세요the section called “보안 모범 사례”.

데이터 삭제

AWS Security Agent에서 데이터를 삭제하는 경우:

- 데이터는 삭제 대상으로 표시되며 서비스를 통해 더 이상 액세스할 수 없습니다.
- 30일 이내에 모든 AWS Security Agent 시스템에서 데이터가 삭제됩니다.

데이터를 삭제하려면

1. AWS 콘솔에서 AWS Security Agent로 이동합니다.
2. 삭제할 데이터(보안 검토, 조사 결과 또는 사용자 지정 요구 사항)를 선택합니다.
3. 삭제를 선택하여 삭제를 확인합니다.

AWS Security Agent의 자격 증명 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 AWS Security Agent 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 AWS 서비스 있는 사용자를 제어합니다. IAM 는 추가 비용 없이 사용할 수 있는 입니다.

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 AWS Security Agent에서 수행하는 작업에 따라 다릅니다.

서비스 사용자 - AWS Security Agent 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 AWS Security Agent 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다.

AWS Security Agent의 기능에 액세스할 수 없는 경우 섹션을 참조하세요the section called “AWS Security Agent 자격 증명 및 액세스 문제 해결”.

서비스 관리자 - 회사에서 AWS Security Agent 리소스를 책임지고 있는 경우 AWS Security Agent에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS Security Agent 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여의 기본 개념을 이해합니다 IAM. 회사가 AWS Security Agent IAM 에서를 사용하는 방법에 대한 자세한 내용은 섹션을 참조하세요the section called “AWS Security Agent의 작동 방식 IAM”.

IAM 관리자 - IAM 관리자인 경우 AWS Security Agent에 대한 액세스를 관리하는 정책을 작성하는 방법에 대한 세부 정보를 알고 싶을 수 있습니다. 에서 사용할 수 있는 AWS Security Agent 자격 증명 기

반 정책 예제를 보려면 섹션을 IAM참조하세요the section called “AWS Security Agent 자격 증명 기반 정책 예제”.

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. IAM 역할을 수임하여 AWS 계정 루트 사용자 IAM 사용자, 또는 로 인증(로그인 AWS)되어야 합니다. AWS는 IAM 역할을 사용하고 루트 사용자를 직접 사용하지 않을 것을 제안합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 자격 증명 AWS 으로에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션 ID로 로그인하면 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법을 AWS참조하세요. https://docs.aws.amazon.com/signin/latest/userguide/how-to-sign-in.html

AWS 프로그래밍 방식으로에 액세스하는 경우는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 AWS 일반 참조의 서명 버전 4 서명 프로세스를 참조하세요.

사용하는 인증 방법에 상관 없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화할 것을 AWS 권장합니다. 자세한 내용은 AWS IAM Identity Center(AWS Single Sign-On 후속) 사용 설명서의 멀티 팩터 인증 및 IAM 사용 설명서의 AWS에서 멀티 팩터 인증(MFA) 사용을 참조하세요.

AWS 계정 루트 사용자

를 처음 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 작업의 전체 목록은 계정 관리 참조 안내서의 루트 사용자 자격 증명이 필요한 작업을 참조하세요.

페더레이션 ID

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, AWS Directory Service, Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스 에 액세스하는 모든 사용자의 사용자입니다. 페더레이션 자격 증명에 액세스할 때 역할을 AWS 계정수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 및 애플리케이션에서 사용할 수 있도록 자체 ID 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 IAM Identity Center란 무엇입니까?를 참조하세요. AWS IAM Identity Center(AWS Single Sign-On 후속) 사용 설명서의 .

IAM 사용자 및 그룹

IAM 사용자 는 단일 사용자 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 그러나 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 IAM 사용자것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 자격 증명이 필요한 사용 사례의 경우, 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 컬렉션을 지정하는 자격 증명입니다 IAM 사용자. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합에 대한 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 IAMAdmins라는 그룹이 있고 해당 그룹에 IAM 리소스를 관리할 수 있는 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 자격 증명을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 (역할 대신) 생성 시기를 참조하세요.

IAM 역할

IAM 역할은 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. 와 비슷 IAM 사용자하지만 특정 사람과는 관련이 없습니다. IAM 역할을 전환 AWS Management Console 하여에서 역할을 일시적으로 수임할 수 있습니다. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-

console.html AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 역할 사용을 참조하세요.

IAM 임시 자격 증명이 있는 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 서드 파티 ID 공급자의 역할 만들기를 참조하세요. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 관리하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관 짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center(AWS Single Sign-On 후속) 사용 설명서의 권한 세트를 참조하세요. AWS Single Sign-On
- 임시 IAM 사용자 권한 -는 IAM 역할을 수임하여 특정 작업에 대해 다른 권한을 일시적으로 수임할 IAM 사용자 수 있습니다.
- 교차 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 교차 계정 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 에서는 정책을 리소스에 직접 연결할 AWS 서비스수 있습니다(역할을 프록시로 사용하는 대신). 교차 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할이 리소스 기반 정책과 어떻게 다른지 참조하세요.
- 교차 서비스 액세스 - 일부는 다른에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어 서비스에서 호출할 때 해당 서비스가에서 애플리케이션을 실행 Amazon EC2 하거나에 객체를 저장하는 것이 일반적입니다 Amazon S3. 서비스는 직접 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 보안 주체 권한 - IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 정책은 보안 주체에게 권한을 부여합니다. 일부 서비스를 사용할 때는 다른 서비스에서 다른 작업을 트리거하는 작업을 수행할 수 있습니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다 IAM. 자세한 정보는 IAM 사용 설명서에서 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.
  - 서비스 연결 역할 - 서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 태스크를 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.
- 에서 Amazon EC2 실행되는 애플리케이션 - IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니

다. 이는 Amazon EC2 인스턴스 내에 액세스 키를 저장하는 것보다 더 좋습니다. Amazon EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 Amazon EC2 인스턴스에서 실행 중인 프로그램이 임시 자격 증명을 가져올 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

IAM 역할을 사용할지 여부를 알아보려면 IAM 사용 설명서의 IAM 역할 생성 시기(사용자 대신)를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는의 객체입니다.는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

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.

모든 IAM 엔터티(사용자 또는 역할)는 권한 없이 시작됩니다. 기본적으로 사용자는 아무 작업도 할 수 없으며 자신의 암호를 변경할 수도 없습니다. 사용자에게 태스크를 수행할 권한을 부여하기 위해 관리자는 사용자에게 권한 정책을 연결해야 합니다. 또한 관리자는 의도한 권한을 가지고 있는 그룹에 사용자를 추가할 수 있습니다. 관리자가 그룹에 권한을 부여하면 해당 그룹의 모든 사용자에게 해당 권한이 부여됩니다.

IAM 정책은 작업을 수행하는 데 사용하는 방법에 관계없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

자격 증명 기반 정책은 IAM 사용자자격 증명, 역할 또는 그룹과 같은 자격 증명에 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 설명서의 IAM 정책 생성을 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은의 여러 사용자, 그룹 및 역할에 연결할 수 있는

독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 Amazon S3 버킷과 같은 리소스에 연결하는 JSON 정책 문서입니다. 서비스 관리자는 이러한 정책을 사용하여 지정된 보안 주체(계정 멤버, 사용자 또는 역할)가 해당 리소스에 대해 수행할 수 있는 작업과 어떤 조건에서 수행할 수 있는지를 정의할 수 있습니다. 리소스 기반 정책은 인라인 정책입니다. 관리형 리소스 기반 정책은 없습니다.

액세스 제어 목록(ACL)

ACL(액세스 제어 목록)은 리소스에 액세스할 수 있는 권한을 가진 보안 주체(계정 멤버, 사용자 또는 역할)를 제어하는 정책의 유형입니다. ACLs은 리소스 기반 정책과 유사하지만 JSON 정책 문서 형식을 사용하지 않습니다. Amazon S3 AWS WAF, 및 Amazon VPC 는 ACLs. ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 참조하십시오.

기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 자격 증명 기반 정책이 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔터티의 ID 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책의 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCPs) - SCPs는의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책입니다 AWS Organizations. AWS Organizations 는 기업이 소유한 여러 AWS 계정 을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자를 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCPs에 대한 자세한 내용은 AWS Organizations 사용 설명서의 SCPs 작동 방식을 참조하세요. AWS Organizations
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 자격 증명 기반 정책과 세션 정책의 교집합입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도

있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 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.

AWS Security Agent의 작동 방식 IAM

IAM 를 사용하여 AWS Security Agent에 대한 액세스를 관리하기 전에 AWS Security Agent에서 사용할 수 있는 IAM 기능에 대해 알아봅니다.

IAM 특성	AWS Security Agent 지원
the section called “AWS Security Agent에 대한 자격 증명 기반 정책”	예
the section called “AWS Security Agent 내의 리소스 기반 정책”	아니요
the section called “AWS Security Agent에 대한 정책 작업”	예
the section called “AWS Security Agent에 대한 정책 리소스”	부분적
the section called “AWS Security Agent에 사용되는 정책 조건 키”	예
the section called “AWS Security Agent의 액세스 제어 목록(ACLs)”	아니요
the section called “AWS Security Agent를 사용한 ABAC(속성 기반 액세스 제어)”	아니요
the section called “AWS Security Agent에서 임시 자격 증명 사용”	예

IAM 특성	AWS Security Agent 지원
the section called “AWS Security Agent에 대한 전달 액세스 세션”	예
the section called “AWS Security Agent의 서비스 역할”	아니요
the section called “AWS Security Agent의 서비스 연결 역할”	예

AWS Security Agent 및 기타에서 AWS 서비스 작업하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS 서비스 에서 작업하는 IAM 섹션을 IAM참조하세요.

AWS Security Agent에 대한 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 자격 증명 기반 정책을 사용하면 허용되거나 거부된 작업 및 리소스와 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 위탁자가 연결된 사용자 또는 역할에 적용되므로 위탁자를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

AWS Security Agent의 자격 증명 기반 정책 예제

AWS Security Agent 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요the section called “AWS Security Agent 자격 증명 기반 정책 예제”.

AWS Security Agent 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 또는 AWS 서비스가 포함될 수 있습니다.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 위탁자로 지정할 수 있습니다. 리소스 기반 정책에 교차 계정 위탁자를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 보안 주체와 리소스가 서로 다른 AWS 계정에 있는 경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 위탁자에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하세요.

AWS Security Agent에 대한 정책 작업

작업 지원 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

자격 IAM 증명 기반 정책의 `Action` 요소는 정책에 의해 허용되거나 거부될 특정 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 이 작업은 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에서 사용됩니다.

AWS Security Agent의 정책 작업은 작업 앞에 접두사를 사용합니다`securityagent:`. 예를 들어 AWS Security Agent `CreateEnvironment` API 작업을 사용하여 환경을 생성할 수 있는 권한을 부여하려면 해당 정책에 `securityagent:CreateEnvironment` 작업을 포함합니다. 정책 문에는 `Action` 또는 `NotAction` 요소가 포함되어야 합니다. AWS Security Agent는이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 세트를 정의합니다.

명령문 하나에 여러 태스크를 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "securityagent:action1",
      "securityagent:action2"
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `List`라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "securityagent:List*"
```

AWS Security Agent에 대한 정책 리소스

정책 리소스 지원: 부분적

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 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. 문에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 명령문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

일부 AWS Security Agent API 작업은 여러 리소스를 지원합니다. 예를 들어 ListEnvironments API 작업을 호출할 때 여러 환경을 참조할 수 있습니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

예를 들어 AWS Security Agent 환경 리소스의 ARN은 다음과 같습니다.

```
arn:${Partition}:securityagent:${Region}:${Account}:environment/${EnvironmentId}
```

문my-environment-2에서 환경 my-environment-1 및를 지정하려면 다음 예제 ARNs 사용합니다.

```
"Resource": [
         "arn:aws:securityagent:us-east-1:123456789012:environment/my-environment-1",
         "arn:aws:securityagent:us-east-1:123456789012:environment/my-environment-2"
```

특정 계정에 속하는 모든 환경을 지정하려면 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:securityagent:us-east-1:123456789012:environment/*"
```

AWS Security Agent에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 문이 적용되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 `OR` 작업을 사용하여 조건을 AWS 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어 리소스에 IAM 사용자 이름 태그가 지정된 경우에만 리소스에 액세스할 수 있는 IAM 사용자 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS Security Agent는 자체 조건 키 세트를 정의하고 일부 전역 조건 키 사용을 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

AWS Security Agent의 액세스 제어 목록(ACLs)

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.

AWS Security Agent를 사용한 ABAC(속성 기반 액세스 제어)

ABAC 지원(정책의 태그): 아니요

AWS Security Agent에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

일부 AWS 서비스는 임시 자격 증명을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 작업하는 AWS 서비스를 비롯한 추가 정보는 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하세요.

사용자 이름과 암호를 제외한 방법을 사용하여 AWS Management Console에 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 SSO(Single Sign-On) 링크를 사용하여 AWS에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다. 그런 다음 이러한 임시 자격 증명을 사용하여 AWS에 액세스할 수 있습니다. AWS는 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 내용은 IAM의 임시 보안 자격 증명 섹션을 참조하세요.

AWS Security Agent에 대한 전달 액세스 세션

전달 액세스 세션(FAS) 지원: 예

IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 호출하는 보안 주체의 권한을 요청 AWS 서비스와 결합하여 다운스트림 서비스에 요청합니다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용을 완료해야 하는 요청을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

AWS Security Agent의 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 서비스에 권한을 위임할 역할 생성을 참조하세요.

AWS Security Agent의 서비스 연결 역할

서비스 연결 역할 지원: 예

서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 태스크를 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 AWS 계정에 표시되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

AWS Security Agent 자격 증명 기반 정책 예제

기본적으로 IAM 사용자 및 역할에는 AWS Security Agent 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console AWS CLI또는 AWS API를 사용하여 작업을 수행할 수 없

습니다. IAM 관리자는 사용자 및 역할에 필요한 지정된 리소스에 대해 특정 API 작업을 수행할 수 있는 권한을 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 해당 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 JSON 편집기를 사용하여 정책 생성을 참조하세요.

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS Security Agent 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 직무에 대한 AWS 관리형 정책을 참조하세요.
- 최소 권한 적용 - IAM 정책으로 권한을 설정할 때 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. 를 사용하여 권한을 적용하는 IAM 방법에 대한 자세한 내용은 IAM 사용 설명서의 의 정책 및 권한을 IAM 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 Analyzer 를 사용하여 IAM 정책을 검증하여 안전하고 기능적인 권한을 보장합니다. 는 정책이 정책 IAM 언어(JSON) 및 IAM 모범 사례를 준수하도록 신규 및 기존 정책을 IAM Access Analyzer 검증합니다.는 안전하고 기능적인 정책을 작성하는 데 도움이 되는 100개 이상의 정책 확인 및 실행 가능한 권장 사항을 IAM Access Analyzer 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 - 계정의 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 켭니다. API 작업을 직접적으로 직접 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 MFA 보호 API 액세스 구성을 참조하세요.

AWS Security Agent 콘솔 사용

AWS Security Agent 콘솔에 액세스하려면 IAM 보안 주체에 최소 권한 집합이 있어야 합니다. 이러한 권한은 보안 주체가의 AWS Security Agent 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 보안 인증 정보 기반 정책을 만들면 콘솔이 해당 정책이 연결된 보안 주체에 대해 의도대로 작동하지 않습니다.

IAM 보안 주체가 AWS Security Agent 콘솔을 계속 사용할 수 있도록 하려면 고유한 이름으로 정책을 생성합니다. 정책을 보안 주체에게 연결하세요. 자세한 내용은 IAM 사용자 설명서의 사용자에게 권한 추가를 참조하세요.

정책 세부 정보는 섹션을 참조하세요the section called “AWS 관리형 정책: SecurityAgentWebAppAPIPolicy”.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

AWS Security Agent 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 AWS Security Agent 및 작업 시 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다 IAM.

AccessDeniedException

AWS API 작업을 호출할 `AccessDeniedException` 때를 수신하면 사용 중인 IAM 보안 주체 자격 증명에 해당 호출에 필요한 권한이 없는 것입니다.

```
An error occurred (AccessDeniedException) when calling the CreateEnvironment operation:
User: arn:aws:iam::111122223333:user/user_name is not authorized to perform:
securityagent:CreateEnvironment on resource:
 arn:aws:securityagent:region:111122223333:environment/my-env
```

이전 예제 메시지에서 사용자는 AWS Security Agent `CreateEnvironment` API 작업을 호출할 권한이 없습니다. IAM 보안 주체에 AWS Security Agent 관리자 권한을 제공하려면 섹션을 참조하세요the section called “AWS Security Agent 자격 증명 기반 정책 예제”.

IAM에 대한 자세한 내용은 IAM 사용 설명서의 정책을 사용하여 AWS 리소스에 대한 액세스 제어를 참조하세요.

내 외부의 사람이 내 AWS Security Agent 리소스에 액세스 AWS 계정 하도록 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.

- AWS Security Agent가 이러한 기능을 지원하는지 여부를 알아보려면 섹션을 참조하세요the section called “AWS Security Agent의 작동 방식 IAM”.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유 AWS 계정 한 다른 IAM 사용자 의에 대한 액세스 권한 제공을 참조하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- 자격 증명 연동을 통해 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부 인증 사용자에게 액세스 권한 제공(자격 증명 연동)을 참조하세요.
- 교차 계정 액세스를 위한 역할 및 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할이 리소스 기반 정책과 어떻게 다른지 참조하세요.

인시던트 대응

AWS Security Agent는 취약성이 악용되기 전에 이를 식별하고 방지하도록 설계된 선제적 보안 테스트 서비스입니다. 이 서비스는 사후 대응 인시던트 탐지 또는 대응 대신 설계 검토, 코드 분석 및 침투 테스트를 통한 예방 보안 검증에 중점을 둡니다.

인시던트 탐지

AWS Security Agent는 인시던트 감지 기능을 제공하지 않습니다. 이 서비스는 개발 중 및 배포 전에 애플리케이션의 취약성을 검증하는 선제적 보안 테스트 도구로 작동합니다. 런타임 보안 모니터링 및 인시던트 감지를 위해 Amazon GuardDuty, AWS Security Hub 또는 Amazon CloudWatch와 같은 서비스를 사용합니다.

인시던트 알림

AWS Security Agent는 보안 인시던트에 대한 실시간 알림을 생성하지 않습니다. 서비스는 설계 검토, 코드 분석 또는 침투 테스트 참여를 완료한 후 AWS 콘솔을 통해 보안 조사 결과를 제공합니다. 이러한 결과는 활성 보안 인시던트가 아닌 테스트 중에 발견된 잠재적 취약성을 나타냅니다.

인시던트 해결

AWS Security Agent는 자동화된 인시던트 수정을 제공하지 않습니다. 서비스는 보안 취약성을 식별하고 다음을 포함한 해결 지침을 제공합니다.

- 식별된 취약성에 대한 자세한 설명
- 검증된 조사 결과를 위한 재현 가능한 악용 경로
- 특정 코드 수정 및 구현 지침
- 발견된 문제에 대한 영향 분석

개발 및 보안 팀은이 지침을 사용하여 취약성이 프로덕션 환경에 도달하기 전에 수동으로 해결합니다.

인시던트 대응 활동 지원

AWS Security Agent는 인시던트 대응을 위해 설계되지 않았지만 보안 팀은 서비스를 사용하여 인시던트 후 활동을 지원할 수 있습니다.

취약성 검증

보안 인시던트 발생 후 AWS Security Agent를 사용하여 유사한 취약성이 다른 애플리케이션 또는 환경에 존재하는지 테스트합니다.

보안 태세 평가

침투 테스트를 수행하여 인시던트 수정의 일부로 구현된 보안 개선 사항을 검증합니다.

근본 원인 분석

코드 보안 검토 기능을 사용하여 다른 코드베이스에 유사한 취약성이 어떻게 존재할 수 있는지 식별합니다.

AWS Security Agent에 대한 규정 준수 검증

AWS 서비스가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반적인 내용은 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact 트의 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS는 규정 준수에 도움이 되도록 다음 리소스를 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모든 AWS 서비스가 HIPAA 자격이 있는 것은 아닙니다.
- AWS Compliance Resources – 사용자의 업계와 위치에 해당할 수 있는 워크북 및 안내서 모음입니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드는 AWS 서비스를 보호하기 위한 모범 사례를 요약하고 여러 프레임워크(미국 국립 표준 기술 연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제 표준화 기구(ISO) 등)의 보안 제어에 지침을 매핑합니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 - AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub -이 AWS 서비스는 AWS 내 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty -이 AWS 서비스는 환경에서 의심스럽고 악의적인 활동을 모니터링하여 AWS 계정, 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager — 이 AWS 서비스는 AWS 사용을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

AWS Security Agent의 복원력

> **Note**
>
> AWS Security Agent는 미국 동부(버지니아 북부) – `us-east-1`, 미국 서부(오레곤) – `us-west-2`, 아시아 태평양(뭄바이) – `ap-south-1`, 아시아 태평양(싱가포르) – `ap-southeast-1`, 아시아 태평양(시드니) – `ap-southeast-2`, 아시아 태평양(도쿄) – `ap-northeast-1`, 유럽(프랑크푸르트) – `eu-central-1`, 유럽(아일랜드) – `eu-west-1`, 남아메리카(상파울루) – 리전에서 사용할 수 있습니다`sa-east-1`. 교차 리전 추론을 사용합니다. 아시아 태평양(뭄바이), 아시아 태평양(싱가포르) 및 남아메리카(상파울루)에서 AWS Security

Agent는 추론 요청을 모든 상용 AWS 리전으로 라우팅하는 글로벌 리전 간 추론을 사용합니다. 다른 모든 리전에서는 지리적 교차 리전 추론을 사용하여 특정 지리적 경계 내에서 데이터 처리를 유지합니다. AWS Security Agent는 애플리케이션 개발 중에 사용되는 도구이므로 중요 인프라 또는 고객 대면 인프라로 배포해서는 안 됩니다.

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 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.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하십시오.

AWS Security Agent의 인프라 보안

관리형 서비스인 AWS Security Agent는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및 AWS가 인프라를 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 AWS Well-Architected Framework의 보안을 참조하세요.

네트워크 격리

AWS Security Agent는 AWS 콘솔 및 AWS Security Agent 웹 애플리케이션을 통해 액세스하는 완전 관리형 서비스입니다. 서비스에 대한 액세스는 ID 제공업체와 통합할 수 있는 AWS Identity and Access Management(IAM) 또는 AWS IAM Identity Center를 통해 제어됩니다.

AWS Security Agent는 AWS PrivateLink로 구동되는 인터페이스 VPC 엔드포인트를 지원하므로 퍼블릭 인터넷을 통과하지 않고도 VPC에서 서비스 APIs에 비공개로 액세스할 수 있습니다. 자세한 내용은 the section called “인터페이스 VPC 엔드포인트” 단원을 참조하십시오.

AWS Security Agent는 대상 애플리케이션에서 침투 테스트를 수행하고 컨트롤 플레인 작업을 수행하려면 인터넷 액세스가 필요합니다. 서비스는 AWS 관리형 인프라를 테스트에 사용하며 계정에서 EC2 인스턴스 또는 기타 컴퓨팅 리소스를 프로비저닝하지 않습니다. 침투 테스트를 위해 VPC 액세스를 구성하는 경우 Security Agent는 서브넷에 ENI를 생성하지만 이 ENI에는 퍼블릭 IP 주소가 없습니다. 자세한 내용은 the section called “에이전트를 프라이빗 VPC 리소스에 연결” 단원을 참조하십시오.

다중 테넌시 및 리소스 격리

AWS Security Agent는 멀티 테넌트 서비스입니다. 보안 검토, 조사 결과 및 고객 데이터는 개별 AWS 계정으로 격리되고 저장 시 암호화됩니다. AWS는 표준 인프라 격리 제어를 적용하여 한 고객의 보안 테스트 활동이 다른 고객의 성능 또는 기밀성에 영향을 미치지 않도록 합니다.

AWS Security Agent 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

AWS PrivateLink를 사용하여 VPC와 AWS Security Agent 간에 프라이빗 연결을 생성할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 Security Agent에 액세스할 수 있습니다. VPC의 인스턴스는 Security Agent에 액세스하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

이 프라이빗 연결은 AWS PrivateLink로 구동되는 인터페이스 엔드포인트를 생성하여 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 Security Agent로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은 PrivateLink 가이드의 Access AWS services AWS PrivateLink throughPrivateLink를 참조하세요. AWS PrivateLink

보안 에이전트에 대한 고려 사항

Security Agent에 대한 인터페이스 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 고려 사항을 검토하세요.

Security Agent는 에이전트 공간 관리 작업, 침투 테스트 및 보안 조사 결과를 포함하여 VPC에서 모든 API 작업을 호출할 수 있도록 지원합니다.

엔드포인트를 생성할 때 IPv4, IPv6 또는 듀얼 스택을 선택할 수 있습니다.

VPC 엔드포인트 정책은 Security Agent에서 지원됩니다. 기본적으로 인터페이스 엔드포인트를 통해 Security Agent에 대한 전체 액세스가 허용됩니다. 엔드포인트 정책을 인터페이스 엔드포인트에 연결하거나 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 액세스를 제어할 수 있습니다.

Security Agent에 대한 모든 API 호출은 VPC 엔드포인트를 통한 호출을 포함하여 TLS 1.2 이상을 사용하여 암호화됩니다. 데이터 보호에 대한 자세한 내용은 the section called “데이터 보호” 섹션을 참조하세요. Security Agent API 호출은 AWS CloudTrail에 로깅됩니다. 자세한 내용은 the section called “예: AWS Security Agent 로그 파일 항목” 단원을 참조하십시오.

Security Agent에 대한 인터페이스 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS 명령줄 인터페이스(AWS CLI)를 사용하여 Security Agent에 대한 인터페이스 엔드포인트를 생성할 수 있습니다. 자세한 내용은 AWS PrivateLink 가이드의 인터페이스 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 Security Agent에 대한 인터페이스 엔드포인트를 생성합니다.

```
com.amazonaws.<region>.securityagent
```

AWS CLI를 사용하여 인터페이스 엔드포인트를 생성하려면 다음 명령을 실행합니다. 리전, VPC ID, 서브넷 IDs 및 보안 그룹 ID를 자체 값으로 바꿉니다.

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --vpc-endpoint-type Interface \
    --service-name com.amazonaws.<region>.securityagent \
    --subnet-ids subnet-1a2b3c4d subnet-5e6f7a8b \
    --security-group-ids sg-1a2b3c4d \
    --private-dns-enabled
```

> **Important**
>
> 인터페이스 엔드포인트에 대해 프라이빗 DNS를 활성화해야 합니다. Security Agent에서는 기본 리전 DNS 이름(예: `securityagent.us-east-1.api.aws`)을 사용하여 엔드포인트를 통해 API를 요청해야 합니다. 엔드포인트별 VPCE URL을 직접 호출하는 것은 지원되지 않습니다.

프라이빗 DNS를 사용하려면 VPC에 `true` 대해 `enableDnsSupport` 및 `enableDnsHostnames` 속성을 모두 로 설정해야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요.

인터페이스 엔드포인트에 대한 보안 그룹 구성

인터페이스 엔드포인트를 생성할 때 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 Security Agent에 대한 트래픽을 제어할 수 있습니다. Security Agent와 통신해야 하는 VPC의 리소스에서 인바운드 HTTPS 트래픽(포트 443)을 허용하는 보안 그룹을 생성합니다.

보안 그룹에는 다음과 같은 인바운드 규칙이 포함되어야 합니다.

- 유형: HTTPS
- 프로토콜: TCP
- 포트 범위: 443
- 소스: VPC CIDR 블록(예: `10.0.0.0/16`) 또는 Security Agent에 액세스해야 하는 리소스의 보안 그룹 IDs

자세한 내용은 AWS PrivateLink 가이드의 보안 그룹을 참조하세요.

엔드포인트의 엔드포인트 정책 생성

엔드포인트 정책은 인터페이스 엔드포인트에 연결할 수 있는 IAM 리소스입니다. 기본 엔드포인트 정책은 인터페이스 엔드포인트를 통해 Security Agent에 대한 전체 액세스를 허용합니다. VPC에서 Security Agent에 허용되는 액세스를 제어하려면 인터페이스 엔드포인트에 사용자 지정 엔드포인트 정책을 연결합니다.

엔드포인트 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체(AWS 계정, IAM 사용자 및 IAM 역할).
- 수행할 수 있는 작업
- 작업을 수행할 수 있는 리소스.

자세한 내용은 AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

예: AWS Security Agent 작업에 대한 VPC 엔드포인트 정책

다음은 AWS Security Agent에 대한 엔드포인트 정책의 예입니다. 엔드포인트에 연결되면이 정책은 모든 리소스의 모든 보안 주체에 대해 나열된 AWS Security Agent 작업에 대한 액세스 권한을 부여합니다.

```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "securityagent:CreatePentest",
            "securityagent:ListPentests",
            "securityagent:BatchGetPentests",
```

```
            "securityagent:StartPentestExecution",
            "securityagent:StopPentestExecution",
            "securityagent:ListFindings",
            "securityagent:BatchGetFindings"
         ],
         "Resource": "*"
      }
   ]
}
```

예: 지정된 AWS 계정의 모든 액세스를 거부하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 엔드포인트를 사용하여 리소스에 대한 123456789012 모든 액세스를 AWS 거부합니다. 이 정책은 다른 계정의 모든 작업을 허용합니다.

```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "securityagent:*",
         "Resource": "*"
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": "123456789012"
         },
         "Action": "securityagent:*",
         "Resource": "*"
      }
   ]
}
```

문제 해결

Security Agent API를 호출할 때 연결 시간 초과

- VPC 엔드포인트 상태가 인지 확인합니다. `available`

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1a2b3c4d \
    --query "VpcEndpoints[].State"
```

- 엔드포인트와 연결된 보안 그룹이 VPC CIDR 또는 소스 보안 그룹에서 포트 443의 인바운드 TCP 트래픽을 허용하는지 확인합니다.
- VPC 라우팅 테이블이 Security Agent 트래픽을 엔드포인트 대신 인터넷 게이트웨이 또는 NAT 게이트웨이로 라우팅하지 않는지 확인합니다.

에 대한 DNS 확인 실패 **`securityagent.<region>.api.aws`**

- 엔드포인트에서 프라이빗 DNS가 활성화되어 있는지 확인합니다.

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1a2b3c4d \
    --query "VpcEndpoints[].PrivateDnsEnabled"
```

- VPC`true`에서 `enableDnsSupport` 및 `enableDnsHostnames`가 모두 로 설정되어 있는지 확인합니다.

```
aws ec2 describe-vpc-attribute --vpc-id vpc-1a2b3c4d --attribute enableDnsSupport
aws ec2 describe-vpc-attribute --vpc-id vpc-1a2b3c4d --attribute enableDnsHostnames
```

`Access denied` Security Agent APIs 호출 시 오류

- VPC 엔드포인트 정책을 확인하여 수행하려는 작업을 허용하는지 확인합니다.
- IAM 자격 증명 정책이 필요한 `securityagent:` 권한을 부여하는지 확인합니다.
- 사용자 지정 엔드포인트 정책을 사용하는 경우 엔드포인트 정책과 IAM 정책이 모두 요청을 허용하는지 확인합니다.

AWS Security Agent의 구성 및 취약성 분석

구성 및 IT 제어는 AWS와 고객 간의 공동 책임입니다. 자세한 내용은 AWS 공동 책임 모델을 참조하십시오.

교차 서비스 혼동된 대리자 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. AWS에서는 교차 서비스 가장으로 인해 혼동된 대리자 문제가 발생할 수 있습니다. 교차 서비스 가장은 한 서비스(직접 호출하는 서비스)가 다른 서비스(직접 호출되는 서비스)를 직접 호출할 때 발생할 수 있습니다. 직접적으로 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하

기 위해 AWS는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 통해 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 교차 서비스 혼동된 대리자 방지를 참조하세요.

AWS Security Agent의 보안 모범 사례

AWS Security Agent는 자체 보안 정책을 개발하고 구현할 때 고려해야 할 여러 보안 기능을 제공합니다. 다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 사용자의 환경에 적절하지 않거나 충분하지 않을 수 있으므로 규정이 아닌 참고용으로만 사용하세요.

침투 테스트를 위해 비프로덕션 환경 사용

AWS Security Agent는 Kali Linux 배포판의 포괄적인 침투 테스트 도구 제품군을 사용합니다. 이러한 도구는 보안 취약성을 식별하도록 설계되었으며 애플리케이션 상태, 데이터 또는 시스템 구성을 수정하는 작업을 수행할 수 있습니다.

모범 사례: 프로덕션 설정을 미러링하는 비프로덕션 환경에 대해 침투 테스트를 수행합니다. 이러한 테스트 환경은 다음과 같아야 합니다.

- 실시간 고객 데이터 또는 민감한 프로덕션 정보를 포함하지 않습니다.
- 프로덕션 시스템에서 격리
- 프로덕션과 유사한 구성 및 보안 제어 기능 보유
- 프로덕션 시스템에 액세스할 수 있는 자격 증명을 사용하지 않습니다.

프로덕션 환경에서 테스트하면 다음과 같은 결과가 발생할 수 있습니다.

- 데이터 수정 또는 삭제
- 서비스 중단 또는 성능 저하
- 의도하지 않은 상태 변경
- 보안 알림 또는 인시던트 대응 절차 트리거

AI 생성 보안 조사 결과 검증

AWS Security Agent는 AI 에이전트를 사용하여 보안 분석을 수행합니다. AI 시스템의 비결정적 특성으로 인해 침투 테스트 실행은 여러 실행에 걸쳐 다양한 결과를 생성할 수 있습니다.

모범 사례: 문제 해결 조치를 취하기 전에 보안 조사 결과를 검증합니다.

- 각 결과에 대해 AWS Security Agent에서 생성한 확인 스크립트 검토
- 테스트 환경에서 확인 스크립트를 실행하여 취약성 확인
- 포괄적인 적용 범위를 보장하기 위해 여러 침투 테스트 실행 고려
- 전문적인 보안 판단을 적용하여 결과 심각도 및 악용 가능성 평가

식별된 모든 문제가 특정 배포 컨텍스트에서 악용 가능한 취약성을 나타낼 수 있는 것은 아닙니다.

생성된 문제 해결 코드 검토 및 테스트

AWS Security Agent는 식별된 취약성에 대한 코드 수정 및 보안 개선을 생성할 수 있습니다. 이러한 AI 생성 수정은 배포 전에 확인이 필요합니다.

모범 사례: 생성된 모든 코드 변경 사항 검토:

- 제안된 수정 사항의 완전성 및 정확성 검사
- 비프로덕션 환경에서 수정 사항을 철저히 테스트합니다.
- 수정으로 인해 새로운 취약성이나 중단 기능이 발생하지 않는지 확인
- AWS Security Agent를 사용하여 수정 사항을 적용한 후 다시 테스트하거나 제공된 확인 스크립트를 실행합니다.
- 조직의 코드 검토 및 승인 프로세스 준수

코드 리포지토리 액세스

AWS Security Agent는 풀 요청 설명 및 코드 검토 통합을 통해 코드 변경에 대한 보안 지침을 제공할 수 있습니다. 민감한 보안 정보를 보호하기 위해이 기능은 특정 제약 조건에서 작동합니다.

제한: 코드 보안 지침은 프라이빗 리포지토리로만 제한됩니다. 이렇게 하면 다음이 보장됩니다.

- 보안 조사 결과는 조직의 기밀을 유지합니다.
- 문제 해결 전에 잠재적 취약성이 공개적으로 공개되지 않음
- 생성된 수정 권장 사항은 악용 세부 정보를 노출하지 않습니다.

AWS Security Agent는 퍼블릭 리포지토리에 대한 코드 보안 지침을 제공하지 않습니다. 보안 조사 결과가 공개적으로 표시되는 퍼블릭 리포지토리 또는 오픈 소스 프로젝트에는 설명하지 않습니다.

AWS Security Agent 침투 테스트는 pentest에 대해 구성한 프라이빗 및 퍼블릭 리포지토리를 검토하고 수정할 수 있습니다. 리포지토리가 퍼블릭인 경우 수정 코드는 풀 요청 대신 다운로드 가능한 diff 파일로 제공됩니다.

액세스 가능한 URLs

액세스 가능한 URLs 테스트 중에 침투 테스트 환경에서 액세스할 수 있는 추가 엔드포인트를 지정합니다. 이는 애플리케이션이 타사 인증 공급자 또는 CDNs과 같은 외부 서비스에 의존하는 경우에 필요합니다. 테스트에 필요한 모든 네트워크 종속성은 대상 URLs 또는 액세스 가능한 URLs로 지정해야 합니다. 네트워크는 지정되지 않은 엔드포인트에 대한 액세스를 차단합니다.

보안 영향: AWS Security Agent는 액세스 가능한 URLs. 액세스 가능한 URLs 지정하면 이러한 종속성에 대한 신뢰를 나타냅니다. 자격 증명을 포함한 침투 테스트 데이터는 테스트 중에 액세스 가능한 URL 엔드포인트로 전송될 수 있습니다.

교차 리전 추론

AWS Security Agent는 추론 요청을 처리할 최적의 리전을 자동으로 선택합니다. 이렇게 하면 사용 가능한 컴퓨팅 리소스와 모델 가용성이 극대화되고 최상의 고객 경험이 제공됩니다. 데이터는 요청이 시작된 리전에만 저장되지만 입력 프롬프트와 출력 결과는 해당 리전 외부에서 처리될 수 있습니다. AWS 네트워크를 통해 암호화된 모든 데이터를 전송합니다.

AWS Security Agent는 리전에 따라 두 가지 유형의 리전 간 추론을 사용합니다.

- 지리적 리전 간 추론 - 대부분의 기능에 대해 특정 지리적 경계(예: 미국, EU, 호주 또는 일본) 내에서 데이터 처리를 유지합니다. 코드 수정의 경우 호주와 일본의 요청은 유럽 연합에서 처리됩니다. 미국 동부(버지니아 북부) – `us-east-1`, 미국 서부(오레곤) – `us-west-2`, 아시아 태평양(시드니) – `ap-southeast-2`, 아시아 태평양(도쿄) – `ap-northeast-1`, 유럽(프랑크푸르트) – `eu-central-1`, 유럽(아일랜드) –에서 사용됩니다`eu-west-1`.
- 글로벌 리전 간 추론 - 모든 상용 AWS 리전에 추론 요청을 라우팅하여 사용 가능한 리소스를 최적화하고 모델 처리량을 높입니다. 아시아 태평양(뭄바이) - `ap-south-1`, 아시아 태평양(싱가포르) - 및 `ap-southeast-1`남아메리카(상파울루) -에서 사용됩니다`sa-east-1`.

글로벌 리전 간 추론을 사용하는 리전의 경우 모든 상용 AWS 리전에서 입력 프롬프트 및 출력 결과가 처리될 수 있습니다. 리전 간 작업 중에 전송되는 모든 데이터는 AWS 네트워크에 남아 있으며 퍼블릭 인터넷을 통과하지 않습니다. AWS 리전 간에 전송 중인 데이터를 암호화합니다.

다음 표에서는 요청이 시작된 리전과 사용된 기능을 기반으로 추론 요청이 처리되는 위치를 설명합니다.

요청 오리진	코드 수정을 제외한 모든 기능	코드 수정
미국 - 미국 동부(버지니아 북부) - us-east-1 , 미국 서부(오레곤) - us-west-2	미국	미국
유럽 연합 - 유럽(아일랜드) - eu-west-1 , 유럽(프랑크푸르트) - eu-central-1	유럽 연합	유럽 연합
호주 - 아시아 태평양(시드니) - ap-southeast-2	호주	유럽 연합
일본 - 아시아 태평양(도쿄) - ap-northeast-1	일본	유럽 연합
남아메리카 - 남아메리카(상파울루) - sa-east-1	모든 상용 AWS 리전	모든 상용 AWS 리전
인도 - 아시아 태평양(뭄바이) - ap-south-1	모든 상용 AWS 리전	모든 상용 AWS 리전
동남아시아 - 아시아 태평양(싱가포르) - ap-southeast-1	모든 상용 AWS 리전	모든 상용 AWS 리전

교차 리전 추론은 항상 활성화되어 있으며 옵트아웃할 수 없습니다. 교차 리전 추론은 고객 콘텐츠를 특정 리전으로 제한하는 서비스 제어 정책(SCPs) 또는 AWS Control Tower의 고객 정책의 영향을 받지 않습니다. 리전 간 처리 중에 AWS Security Agent가 데이터를 보호하는 방법에 대한 자세한 내용은 리전 간 데이터 처리를 참조하세요.

IAM 작업 및 리소스 마이그레이션

AWS Security Agent는 모든 환경의 개발 수명 주기 동안 애플리케이션을 사전에 보호하는 프론티어 에이전트입니다. 2026년 2월 9일 이전에 AWS Security Agent에 온보딩한 경우 2026년 3월 9일에 기존 에이전트 인스턴스 리소스 및 AWS Security Agent IAM 작업이 변경될 예정입니다. 퍼블릭 API/SDK 지원 릴리스를 준비하기 위해 에이전트 인스턴스 리소스의 이름이 에이전트 공간으로 변경되고 특정 IAM 작업의 이름이 변경됩니다. 이러한 변경 사항은 2026년 3월 9일 이전에 생성한 모든 애플리

케이션 또는 에이전트 인스턴스 IAM 역할에 영향을 미칩니다. 2026년 3월 9일 이후에 인증 문제가 표시되지 않도록 하려면 마이그레이션 준비의 단계를 따라야 합니다.

> **Note**
>
> 2026년 2월 9일 이후에 새 에이전트 인스턴스를 생성하면 새 에이전트 인스턴스가 에이전트 공간으로 생성되므로 마이그레이션 단계가 필요하지 않습니다.

계획된 변경 사항

AWS Security Agent에서 에이전트 인스턴스 리소스의 이름을 에이전트 공간으로 바꾸고 있습니다. 이름이 로 `arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*` 변경되었습니다`arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*`. 또한 다음 IAM 작업의 이름이 변경됩니다.

- `securityagent:ListAgentInstances`의 이름이 `securityagent:ListAgentSpaces`로 변경됨
- `securityagent:ListControls`의 이름이 `securityagent:ListSecurityRequirements`로 변경됨
- `securityagent:BatchGetAgentInstances`의 이름이 `securityagent:BatchGetAgentSpaces`로 변경됨
- `securityagent:BatchGetSecurityTestContentMetadata`의 이름이 `securityagent:BatchGetPentestJobContentMetadata`로 변경됨
- `securityagent:BatchGetTasks`의 이름이 `securityagent:BatchGetPentestJobTasks`로 변경됨
- `securityagent:CreateDocumentReview`의 이름이 `securityagent:CreateDesignReview`로 변경됨
- `securityagent:GetDocumentReview`의 이름이 `securityagent:GetDesignReview`로 변경됨
- `securityagent:GetDocumentReviewArtifact`의 이름이 `securityagent:GetDesignReviewArtifact`로 변경됨
- `securityagent:ListDocumentReviews`의 이름이 `securityagent:ListDesignReviews`로 변경됨
- `securityagent:ListDocumentReviewComments`의 이름이 `securityagent:ListDesignReviewComments`로 변경됨

- securityagent:ListTasks의 이름이 securityagent:ListPentestJobTasks로 변경됨
- securityagent:StartPentestExecution의 이름이 securityagent:StartPentestJob로 변경됨
- securityagent:StopPentestExecution의 이름이 securityagent:StopPentestJob로 변경됨
- securityagent:DeleteDocumentReview의 이름이 securityagent:DeleteDesignReview로 변경됨

마이그레이션 준비

2026년 3월 9일 이전에 AWS Security Agent를 계속 사용하면서 2026년 3월 9일 이후에도 문제가 발생하지 않도록 하려면 2026년 3월 9일까지 IAM 역할/정책에서 새 리소스와 이전 리소스 및 IAM 작업을 모두 신뢰해야 합니다. 아래 지침은 새 리소스 및 작업 형식으로 마이그레이션하기 위한 가이드를 제공합니다.

1. AWS 계정에 로그인하고 AWS Security Agent 콘솔로 이동합니다.
2. 왼쪽 패널에서 설정을 선택하고 서비스 역할에서 역할을 선택합니다.
3. 연결된 역할에 대한 IAM 콘솔에서 권한 추가 및 정책 연결을 선택합니다.
4. AWSSecurityAgentWebAppPolicy를 선택하고 권한 추가를 선택합니다.
   a. 중요 참고 사항: SecurityAgentWebAppAPIPolicy가 아닌 AWSSecurityAgentWebAppPolicySecurityAgentWebAppAPIPolicy를 새 정책으로 선택했는지 확인합니다.
5. 권한 정책에서 IAM 역할에 AWSSecurityAgentWebAppPolicy와 SecurityAgentWebAppAPIPolicy가 모두 있는지 확인합니다.
6. 동일한 IAM 역할 콘솔에서 신뢰 관계를 선택한 다음 신뢰 정책 편집을 선택합니다.
7. 신뢰 정책을 다음 형식으로 업데이트하여 {{accountId}}를 AWS 계정 ID로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{accountId}}"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:securityagent:us-east-1:{{accountId}}:application/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:SetContext",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{accountId}}"
        },
        "ForAllValues:ArnEquals": {
          "sts:RequestContextProviders": "arn:aws:iam::aws:contextProvider/
IdentityCenter"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:securityagent:us-east-1:{{accountId}}:application/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*"
          ]
        }
      }
    }
  ]
}
```

8. AWS Security Agent 콘솔로 돌아갑니다. 왼쪽 패널에서 에이전트 스페이스를 선택합니다.

9. 침투 테스트가 활성화된 각 에이전트 공간에 대해 다음 단계를 수행합니다.

a. 에이전트 공간으로 이동하여 침투 테스트를 선택합니다.

b. 서비스 액세스에서 서비스 역할 이름 아래의 역할을 선택합니다.

c. 연결된 역할에 대한 IAM 콘솔에서 신뢰 관계를 선택한 다음 신뢰 정책 편집을 선택합니다.

d. 신뢰 정책을 다음 형식으로 업데이트하여 {{accountId}}를 AWS 계정 ID로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "securityagent.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "{{accountId}}"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*",
            "arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*"
          ]
        }
      }
    }
  ]
}
```

AWS CloudTrail을 사용하여 AWS Security Agent API 호출 로깅

AWS Security Agent는 AWS Security Agent에서 사용자, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 AWS Security Agent에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 AWS Security Agent 콘솔의 호출과 AWS Security Agent API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 AWS Security Agent에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS Security Agent에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

AWS CloudTrail의 Security Agent 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. AWS Security Agent에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

AWS Security Agent에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

모든 AWS Security Agent 작업은 CloudTrail에서 로깅됩니다. 예를 들어 `CreatePentest`, `BatchGetPentestJobs` 및 `ListFindings` 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.

- 요청이 루트 또는 AWS Identity and Access Management(IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

예: AWS Security Agent 로그 파일 항목

AWS Security Agent 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 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출에 대한 순서가 지정된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음은 `CreatePentest` 태스크를 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2025-01-15T10:30:00Z",
    "eventSource": "securityagent.amazonaws.com",
    "eventName": "CreatePentest",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.12",
    "userAgent": "aws-cli/2.13.0",
    "requestParameters": {
        "pentestName": "WebApp-Security-Test",
        "targetUrl": "https://example.com",
        "testScope": "OWASP-Top-10"
    },
    "responseElements": {
        "pentestId": "pt-1234567890abcdef0",
        "pentestArn": "arn:aws:securityagent:us-east-1:123456789012:pentest/
pt-1234567890abcdef0"
    },
    "requestID": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
    "eventID": "12345678-1234-1234-1234-123456789012",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
```

```
    "eventCategory": "Management"
}
```

Service Quotas

AWS Security Agent에는 생성할 수 있는 리소스 수와 작업을 수행할 수 있는 속도를 제한하는 할당량이 있습니다. 조정 가능으로 표시된 할당량은 AWS Support를 통해 요청을 제출하여 늘릴 수 있습니다. 자세한 내용은 지원 사례 및 사례 관리 생성을 참조하세요.

작업 할당량

운영 할당량은 서비스 용량을 관리하는 데 도움이 되는 보안 테스트 및 검토 기능의 월별 사용을 제한합니다.

Resource	Scope	할당량	조정 가능
설계 검토	리전별 계정당 월별	200	예
PR 코드 검토	리전별 계정당 월별	1,000	예

구성 할당량

구성 할당량은 AWS Security Agent 환경에서 구성할 수 있는 리소스 및 설정 수를 제한합니다.

Resource	Scope	할당량	조정 가능
에이전트 공간	리전별 계정당	100	예
통합	리전별 계정당	20	아니요
통합당 통합 리소스	통합당	50	아니요
사용자 지정 보안 요구 사항	리전별 계정당	20	아니요
Pentest 프로젝트	리전별 계정당	1,000	예
동시 Pentest 실행	리전별 계정당	5	예
코드 검토 프로젝트	리전별 계정당	1,000	예

Resource	Scope	할당량	조정 가능
동시 코드 검토 실행	리전별 계정당	5	예

문서 기록

다음 표에서는 AWS Security Agents 사용 설명서의 주요 업데이트 및 새로운 기능에 대해 설명합니다.

변경 사항	설명	날짜
프라이빗 연결(미리 보기)	프라이빗 연결에 대한 설명서가 추가되었습니다. 이 새로운 기능을 통해 AWS Security Agent는 시스템을 퍼블릭 인터넷에 노출하지 않고 Amazon VPC Lattice를 사용하여 프라이빗 네트워크에서 실행되는 소스 제어 시스템에 연결할 수 있습니다.	2026년 6월 16일
위협 모델링(미리 보기)	위협 모델링에 대한 설명서가 추가되었습니다. 이 새로운 기능은 설계 문서(범위 문서), 소스 코드(소스) 또는 둘 다에서 애플리케이션의 위협 모델을 구축합니다. 범위 문서는 분석의 초점을 정의하는 기능 설계 문서인 반면, 소스 코드는 기존 시스템에 대한 컨텍스트를 제공합니다. 범위 문서를 제공하지 않으면 에이전트는 소스 코드에서만 위협 모델을 생성합니다. 각 실행은 시스템 개요와 심각도 등급 및 권장 사항이 포함된 STRIDE 범주로 분류된 위협 세트를 생성합니다.	2026년 6월 8일

전체 리포지토리 코드 검토(미리 보기)	전체 리포지토리 코드 검토를 위한 설명서가 추가되었습니다. 이 새로운 기능은 전체 코드베이스에 대한 컨텍스트 인식 보안 분석을 수행하고 결과에 대한 코드 수정을 생성합니다.	2026년 5월 12일
문제 해결을 위한 리전 가용성 업데이트	AWS Security Agent 웹 앱에서 침투 테스트 결과 수정을 위한 리전 가용성이 업데이트되었습니다.	2026년 4월 16일
조사 결과 수정을 활성화하기 위한 리전 가용성 업데이트	AWS Management Console에서 침투 테스트 결과 수정을 활성화하기 위한 리전 가용성이 업데이트되었습니다.	2026년 4월 16일
고객 관리형 키 지원	고객 관리형 키(CMK) 지원에 대한 설명서가 추가되었습니다. 이제 에이전트 스페이스 및 통합을 생성할 때 고객 관리형 KMS 키를 지정하여 제어하는 키로 데이터를 암호화할 수 있습니다.	2026년 3월 31일
AWS 관리형 정책 업데이트	새로운 TargetDomain 및 DesignReviewFeedback 리소스 유형에 대한 AWSSecurityAgentWebAppPolicy 관리형 정책이 추가되었습니다.	2026년 3월 31일
AWS 관리형 정책 업데이트	새 AgentSpace 리소스 유형 및 IAM 작업 이름 변경에 대한 AWSSecurityAgentWebAppPolicy 관리형 정책이 추가되었습니다.	2026년 2월 9일

AWS 관리형 정책 업데이트	고객이 설계 검토를 삭제할 수 있도록 SecurityAgentWebAppAPIPolicy를 업데이트했습니다.	2026년 1월 28일
AWS 관리형 정책 업데이트	고객이 보안 조사 결과에 대한 자동 코드 수정을 시작할 수 있도록 SecurityAgentWebAppAPIPolicy를 업데이트했습니다.	2026년 1월 20일
AWS 관리형 정책 업데이트	고객이 콘솔에서 이미지를 볼 수 있도록 SecurityAgentWebAppAPIPolicy로 업데이트되었습니다.	2025년 12월 5일
AWS Security Agents 최초 릴리스	서비스 출시에 대한 최초 설명서	2025년 12월 2일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.