Panduan Pengguna

Agen Keamanan AWS

Agen Keamanan AWS: Panduan Pengguna

Table of Contents

Pengantar

Pelajari apa yang dilakukan AWS Security Agent, cara kerjanya, dan bagaimana sumber dayanya cocok.

Apa itu Agen Keamanan AWS?

AWS Security Agent adalah agen perbatasan yang secara proaktif mengamankan aplikasi Anda selama siklus pengembangan. Ini melakukan tinjauan keamanan otomatis yang disesuaikan dengan kebutuhan organisasi Anda, membangun model ancaman untuk mengidentifikasi bagaimana aplikasi Anda dapat diserang, dan memberikan pengujian penetrasi sadar konteks sesuai permintaan. Dengan terus memvalidasi keamanan dari desain hingga penerapan, AWS Security Agent membantu mencegah kerentanan lebih awal di semua lingkungan Anda.

Tim keamanan menentukan persyaratan keamanan organisasi sekali di AWS Console: pustaka otorisasi yang disetujui, standar pencatatan, dan kebijakan akses data. AWS Security Agent secara otomatis memberlakukan persyaratan keamanan ini selama pengembangan, mengevaluasi dokumen arsitektur dan kode sesuai standar Anda dan memberikan panduan khusus ketika mendeteksi pelanggaran. Ini memberikan penegakan keamanan yang konsisten untuk tinjauan desain dan kode di semua tim.

Untuk validasi penerapan, AWS Security Agent mengubah pengujian penetrasi dari bottleneck berkala menjadi kemampuan sesuai permintaan. Tim keamanan menyediakan URL target, detail otentikasi, kode sumber, dan dokumentasi aplikasi. AWS Security Agent mengembangkan pemahaman aplikasi yang mendalam dan mengeksekusi rantai serangan canggih untuk menemukan dan memvalidasi kerentanan, memungkinkan tim untuk menguji kapan pun diperlukan.

Kemampuan kunci

AWS Security Agent menyediakan kemampuan keamanan komprehensif yang mencakup seluruh siklus hidup pengembangan.

Tinjauan keamanan desain

AWS Security Agent mengalihkan keamanan yang tersisa dengan memberikan umpan balik keamanan real-time pada dokumen desain dan menilai kepatuhan terhadap persyaratan keamanan organisasi sebelum kode apa pun ditulis. Tim keamanan mengunggah dokumen melalui aplikasi web

dan menerima panduan remediasi untuk memprioritaskan temuan, mempercepat tinjauan manual yang memakan waktu ke dalam analisis terfokus. Dengan secara proaktif menyematkan standar keamanan Anda ke dalam setiap tinjauan desain, Anda mengurangi pengerjaan ulang arsitektur tahap akhir dan mengikuti beberapa tim pengembangan.

Pemodelan ancaman

AWS Security Agent membuat model ancaman aplikasi Anda untuk mengidentifikasi bagaimana mereka dapat diserang. Berikan dokumen desain teknis (scope docs) untuk menentukan fokus agen, kode sumber sebagai konteks bagi agen untuk memahami sistem Anda yang ada, atau keduanya. AWS Security Agent menghasilkan ikhtisar sistem yang menjelaskan arsitektur, komponen, batas kepercayaan, aliran data, dan postur keamanan aplikasi Anda, bersama dengan serangkaian ancaman yang diklasifikasikan berdasarkan kategori STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) dengan tingkat keparahan dan rekomendasi yang dapat ditindaklanjuti. Model ancaman adalah konfigurasi yang dapat digunakan kembali yang dapat Anda jalankan kembali saat kode dan desain Anda berkembang, memungkinkan penilaian keamanan berulang selama siklus hidup pengembangan.

Tinjauan keamanan kode

AWS Security Agent secara proaktif mengamankan aplikasi melalui dua pendekatan yang saling melengkapi. Pertama, Anda dapat membuat ulasan kode di aplikasi web untuk melakukan pemindaian komprehensif kode sumber lengkap Anda dari GitHub,, Bitbucket GitLab, atau repositori Server GitHub Perusahaan atau bucket S3, mengidentifikasi kerentanan keamanan dan memvalidasi kepatuhan dengan persyaratan organisasi Anda di seluruh basis kode Anda. Kedua, Anda dapat mengaktifkan analisis permintaan tarik otomatis untuk repositori yang terhubung, tempat AWS Security Agent meninjau perubahan kode dan memposting temuan keamanan secara langsung sebagai permintaan tarik atau menggabungkan komentar permintaan. Dalam kedua kasus tersebut, pengembang menerima panduan remediasi, dan AWS Security Agent dapat secara otomatis membuat permintaan tarik atau menggabungkan permintaan dengan perbaikan kode untuk kerentanan yang teridentifikasi. Ini menyematkan keahlian keamanan di semua repositori, mengurangi penundaan terkait keamanan dalam pipeline pengembangan dan evaluasi penskalaan di semua basis kode.

Pengujian penetrasi

AWS Security Agent memberikan pengujian penetrasi sesuai permintaan dengan menggunakan agen AI khusus untuk menemukan, memvalidasi, melaporkan, dan memulihkan kerentanan

keamanan melalui skenario serangan multi-langkah yang disesuaikan. Agen memahami konteks aplikasi Anda dengan menganalisis kode sumber dan dokumentasi untuk mengidentifikasi dan mengeksploitasi kerentanan yang tidak dapat ditemukan oleh alat pemindaian keamanan otomatis. Ini mendokumentasikan temuan dengan analisis dampak, jalur serangan yang dapat direproduksi, dan membuat permintaan tarik dengan perbaikan kode yang siap diterapkan, mengubah penilaian berkala menjadi validasi berkelanjutan yang menskalakan di semua aplikasi daripada terbatas hanya pada yang kritis.

Manfaat

On-demand aksesibilitas

AWS Security Agent menyediakan akses langsung ke keahlian keamanan. Tim pengembangan dapat menjalankan tinjauan desain, model ancaman, analisis kode, dan tes penetrasi sesuai permintaan kapan pun diperlukan, sesuai dengan laju siklus pengembangan modern dan memungkinkan keamanan proaktif di setiap tahap.

Validasi persyaratan organisasi secara otomatis

Tentukan persyaratan keamanan organisasi Anda sekali di AWS Console. AWS Security Agent secara otomatis memvalidasi persyaratan ini di semua aplikasi selama setiap peninjauan desain dan keamanan kode, memastikan tim memenuhi persyaratan spesifik Anda daripada daftar periksa umum.

Skala keahlian keamanan

AWS Security Agent menskalakan tinjauan keamanan agar sesuai dengan kecepatan pengembangan dengan mengotomatiskan penegakan persyaratan keamanan organisasi. Konfigurasikan persyaratan secara terpusat, lakukan tinjauan komprehensif dengan analisis temuan, dan kelola cakupan pengujian penetrasi di seluruh organisasi Anda melalui AWS Console.

Perbaikan yang dapat ditindaklanjuti untuk kerentanan yang dikonfirmasi

AWS Security Agent memvalidasi temuan keamanan yang ditemukan selama pengujian penetrasi melalui eksploitasi berbasis bukti, memberikan jalur eksploitasi yang dapat direproduksi, analisis dampak komprehensif, dan membuat permintaan tarik dengan perbaikan yang siap diterapkan dalam bahasa yang ramah pengembang. Ini membantu tim fokus pada risiko keamanan berdampak tinggi yang sah tanpa membuang waktu untuk positif palsu.

Dukungan cloud multi dan hybrid

AWS Security Agent beroperasi di seluruh lingkungan AWS, on-premise, hybrid, multicloud, dan SaaS, memastikan panduan dan pengujian keamanan yang konsisten terlepas dari pilihan infrastruktur atau platform Anda.

Kemampuan AWS Security Agent

AWS Security Agent menyediakan empat kemampuan keamanan inti di seluruh siklus pengembangan Anda:

- Tinjauan keamanan desain — Meninjau dokumen desain dan arsitektur untuk mengidentifikasi risiko keamanan. Unggah dokumen melalui aplikasi web, dan layanan menganalisisnya terhadap persyaratan keamanan organisasi Anda. AWS Security Agent mengevaluasi kepatuhan terhadap persyaratan keamanan yang Anda tetapkan seperti pustaka otorisasi yang disetujui, standar pencatatan, dan kebijakan akses data. Ini membantu Anda menangkap desain yang tidak aman dan pelanggaran kebijakan di awal proses pengembangan.
- Pemodelan ancaman — Membangun model ancaman aplikasi Anda untuk mengidentifikasi bagaimana hal itu bisa diserang. Berikan dokumen desain sebagai dokumen lingkup untuk menentukan fokus analisis, kode sumber sebagai konteks bagi agen untuk memahami sistem Anda yang ada, atau keduanya. AWS Security Agent menghasilkan ikhtisar sistem yang menjelaskan arsitektur, komponen, batas kepercayaan, aliran data, dan postur keamanan aplikasi Anda, bersama dengan serangkaian ancaman yang diklasifikasikan berdasarkan kategori STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) dengan tingkat keparahan dan rekomendasi yang dapat ditindaklanjuti. Setiap ancaman terhubung kembali ke bukti dalam kode sumber Anda.
- Tinjauan keamanan kode — Menganalisis kode di repositori dan sumber S3 Anda untuk mengidentifikasi kerentanan keamanan dan pelanggaran persyaratan keamanan organisasi di seluruh bahasa, kerangka kerja, dan arsitektur. Buat ulasan kode di aplikasi web untuk melakukan pemindaian komprehensif kode sumber lengkap Anda, atau aktifkan komentar permintaan tarik untuk meninjau perubahan kode secara otomatis. GitHub AWS Security Agent menyediakan panduan remediasi khusus dan dapat secara otomatis menghasilkan permintaan tarik dengan perbaikan kode untuk kerentanan yang teridentifikasi. Ini memastikan penegakan kebijakan keamanan Anda secara konsisten di semua tim pengembangan.
- On-demand pengujian penetrasi - Menemukan, memvalidasi, melaporkan, dan memulihkan kerentanan keamanan dalam aplikasi web langsung dan API melalui skenario serangan multi-

langkah yang disesuaikan. Konfigurasikan layanan untuk membuat pentest melalui aplikasi web dengan menentukan lingkup pengujian, detail otentikasi, dan sumber daya. AWS Security Agent mengembangkan konteks aplikasi dari kode sumber dan dokumentasi yang disediakan dan mengeksekusi rantai serangan canggih untuk mengidentifikasi kerentanan yang dapat dieksploitasi yang dilewatkan oleh analisis statis dan alat konvensional. Ini juga menyediakan perbaikan kode yang siap diterapkan dan membuat permintaan tarik langsung ke repositori kode Anda, memungkinkan Anda untuk menyelesaikan kerentanan lebih cepat.

Cara kerja AWS Security Agent

AWS Security Agent beroperasi di tiga antarmuka untuk menyediakan keamanan aplikasi proaktif selama siklus hidup pengembangan. Konfigurasi keamanan dikelola di AWS Management Console, tinjauan keamanan dilakukan di Aplikasi Web Agen Keamanan, dan perbaikan pencarian kode dan keamanan otomatis terjadi secara langsung di platform repositori kode seperti. GitHub

Gambaran umum

AWS Security Agent terdiri dari tiga komponen utama:

- AWS Management Console - Konfigurasikan Ruang Agen, tentukan persyaratan keamanan, konfigurasikan pengujian penetrasi, sambungkan repositori kode, dan kelola akses pengguna
- Aplikasi Web Agen Keamanan - Melakukan tinjauan desain, membuat dan menjalankan model ancaman, menjalankan tes penetrasi, membuat dan menjalankan ulasan kode, dan meninjau temuan keamanan dalam Ruang Agen yang Anda tetapkan
- Integrasi Repositori Kode - Menerima ulasan kode otomatis pada permintaan tarik dan permintaan tarik remediasi uji penetrasi. Saat ini AWS Security Agent mendukung koneksi ke GitHub.

Anda bekerja dengan AWS Security Agent di salah satu dari tiga peran, yang dapat Anda identifikasi berdasarkan antarmuka yang Anda gunakan:

Peran	Di mana Anda bekerja, dan apa yang Anda lakukan
Administrator	Bekerja di AWS Management Console — menyiapkan Ruang Agen, menentukan persyaratan keamanan, mengonfigurasi kemampuan, dan mengelola akses pengguna.

Peran	Di mana Anda bekerja, dan apa yang Anda lakukan
Pengguna	Bekerja di Aplikasi Web Agen Keamanan — menjalankan tinjauan desain, model ancaman, tes penetrasi, dan ulasan kode, dan meninjau temuan yang dihasilkan.
Pengembang	Bekerja di GitHub atau IDE (seperti Kiro atau Claude Code) — menerima temuan keamanan otomatis pada permintaan tarik dan menjalankan pemindaian kode tanpa meninggalkan lingkungan pengembangan.

Kami menggunakan nama peran ini untuk menunjukkan siapa yang melakukan setiap tugas. Satu orang dapat memegang lebih dari satu peran.

Konfigurasi konsol

Administrator mengonfigurasi AWS Security Agent melalui AWS Management Console.

Agent Spaces - Saat Anda membuat Ruang Agen pertama di AWS Management Console, AWS Security Agent membuat Aplikasi Web untuk akun Anda. Setiap Ruang Agen yang Anda buat mewakili aplikasi atau proyek berbeda yang ingin Anda amankan. Di Aplikasi Web, pengguna memilih Ruang Agen mana yang akan digunakan saat melakukan penilaian keamanan.

Persyaratan keamanan - Tentukan persyaratan keamanan yang dievaluasi AWS Security Agent selama peninjauan desain dan kode, yang disusun ke dalam paket persyaratan keamanan. Aktifkan AWS-managed paket berdasarkan standar industri, buat paket khusus untuk kebijakan organisasi Anda, atau buat persyaratan dengan mengunggah dokumentasi keamanan Anda. Persyaratan berlaku di semua Ruang Agen.

Konfigurasi pengujian penetrasi - Konfigurasikan kemampuan pengujian penetrasi untuk setiap Ruang Agen dengan:

- Memverifikasi domain target untuk pengujian melalui verifikasi DNS atau HTTP
- Mengkonfigurasi akses VPC untuk menguji aplikasi pribadi
- Menyiapkan CloudWatch logging untuk uji penetrasi berjalan
- Mengonfigurasi fungsi AWS Secrets Manager atau Lambda untuk kredensyal pengujian
- Menentukan bucket S3 untuk konteks aplikasi tambahan

Konfigurasi peninjauan kode - Konfigurasikan kemampuan peninjauan kode untuk setiap Ruang Agen dengan:

- Menghubungkan GitHub repositori atau bucket S3 yang berisi kode sumber
- Memilih pengaturan peninjauan kode (kerentanan keamanan, persyaratan khusus, atau keduanya)
- Mengaktifkan komentar permintaan tarik untuk peninjauan otomatis perubahan kode GitHub
- Menyiapkan CloudWatch pencatatan untuk peninjauan kode berjalan

Konfigurasi pemodelan ancaman - Konfigurasikan kemampuan pemodelan ancaman untuk setiap Ruang Agen dengan:

- Menghubungkan repositori kode sumber atau bucket S3 yang berisi kode sumber
- Menambahkan dokumen lingkup (dokumen desain fitur) untuk memfokuskan analisis pada fitur tertentu
- Mengkonfigurasi CloudWatch logging untuk model ancaman berjalan
- Menyiapkan peran layanan untuk akses sumber daya AWS

Integrasi - Hubungkan GitHub repositori ke setiap Ruang Agen untuk mengaktifkan kemampuan utama:

- Menyediakan konteks aplikasi untuk pengujian penetrasi yang lebih akurat
- Aktifkan peninjauan kode untuk pemindaian repositori lengkap dan analisis permintaan tarik
- Aktifkan remediasi kode otomatis melalui permintaan tarik untuk tinjauan kode dan temuan uji penetrasi

Akses pengguna - Kelola cara pengguna mengakses Aplikasi Web Agen Keamanan. Jika Anda telah mengaktifkan IAM Identity Center (SSO), tetapkan pengguna di konsol AWS Security Agent untuk menyediakan akses SSO langsung ke Aplikasi Web. Jika Anda menggunakan IAM-only akses, pengguna dengan akses AWS Console dapat meluncurkan Aplikasi Web melalui tautan akses admin di Konsol untuk Ruang Agen apa pun.

Kegiatan Aplikasi Web

Pengguna mengakses Aplikasi Web Agen Keamanan untuk melakukan penilaian keamanan dalam Ruang Agen yang ditugaskan.

Pilih Ruang Agen - Saat masuk ke Aplikasi Web, pengguna memilih Ruang Agen mana yang akan digunakan. Pengguna hanya dapat melihat dan mengakses Ruang Agen tempat mereka ditugaskan.

Tinjauan desain - Unggah dokumen desain dan spesifikasi arsitektur untuk analisis terhadap persyaratan keamanan organisasi. Tinjau temuan dengan panduan remediasi.

Model ancaman - Buat dan jalankan model ancaman dengan menyediakan kode sumber, dokumen desain teknis (dokumen lingkup), atau keduanya. Dokumen lingkup menentukan apa yang agen fokuskan analisisnya; kode sumber menyediakan konteks tentang sistem Anda yang ada. Setiap proses menghasilkan ikhtisar sistem yang menjelaskan arsitektur aplikasi Anda, batas kepercayaan, aliran data, dan postur keamanan, bersama dengan serangkaian ancaman yang diklasifikasikan menurut kategori STRIDE dengan tingkat keparahan dan rekomendasi yang dapat ditindaklanjuti.

Ulasan kode - Buat dan jalankan ulasan kode yang melakukan analisis statis komprehensif di seluruh kode sumber lengkap Anda. Pilih GitHub repositori atau sumber S3, konfigurasikan pengaturan pemindaian, dan tinjau temuan keamanan terperinci dengan panduan remediasi. AWS Security Agent mengidentifikasi kerentanan keamanan dan memvalidasi kepatuhan terhadap persyaratan keamanan khusus organisasi Anda di seluruh basis kode Anda.

Tes penetrasi - Konfigurasikan dan jalankan tes penetrasi dengan memberikan URL target, detail otentikasi, dan dokumentasi. AWS Security Agent melakukan pengujian otonom untuk menemukan kerentanan yang dapat dieksploitasi melalui skenario serangan multi-langkah.

Tinjau temuan - Periksa temuan keamanan terperinci dari tinjauan desain, model ancaman, tinjauan kode, dan tes penetrasi, termasuk analisis dampak, jalur serangan yang dapat direproduksi, dan panduan remediasi.

Validasi perbaikan - penilaian Re-run keamanan setelah menerapkan remediasi untuk memverifikasi kerentanan telah ditangani.

GitHub integrasi

AWS Security Agent terintegrasi secara langsung GitHub untuk memberikan umpan balik keamanan otomatis dalam alur kerja pengembang.

Tarik komentar permintaan - Setelah administrator menginstal GitHub Aplikasi AWS Security Agent dan mengaktifkan peninjauan kode dengan komentar peninjauan Kode untuk Ruang Agen, AWS Security Agent secara otomatis menganalisis permintaan tarik di repositori yang terhubung. Pengembang menerima temuan keamanan dan panduan remediasi secara langsung dalam komentar permintaan tarik, memvalidasi perubahan kode terhadap persyaratan keamanan organisasi dan kerentanan umum.

Remediasi otomatis - Saat pengguna mengaktifkan remediasi kode otomatis untuk peninjauan kode, AWS Security Agent menghasilkan perbaikan untuk kerentanan yang teridentifikasi dan mengirimkan permintaan tarik ke repositori terkait. GitHub

Remediasi pengujian penetrasi - Saat administrator mengaktifkan pencarian remediasi di Konsol, pengguna dapat meminta remediasi otomatis untuk temuan pengujian penetrasi dari Aplikasi Web. AWS Security Agent membuka permintaan tarik ke GitHub repositori terkait dengan perbaikan kode untuk mengatasi kerentanan.

Memahami hierarki sumber daya dan siklus hidup

AWS Security Agent mengatur sumber daya pengujian keamanan dalam struktur hierarkis yang menentukan apa yang dibagikan di seluruh organisasi Anda dan apa yang dicakup per aplikasi. Memahami struktur ini membantu Anda mengonfigurasi AWS Security Agent secara efektif dan mengetahui di mana menemukan dan mengelola sumber daya yang berbeda.

Apa yang dibagikan di seluruh organisasi Anda

Beberapa sumber daya di AWS Security Agent dikonfigurasi satu kali di tingkat organisasi dan berlaku di semua aplikasi dan Ruang Agen Anda. Sumber daya tingkat penyewa ini memberikan konsistensi dan mengurangi pekerjaan konfigurasi duplikat.

Sumber daya	Apa itu	Mengapa itu dibagikan
Persyaratan keamanan	Standar keamanan organisasi yang menentukan apa yang divalidasi AWS Security Agent selama peninjauan desain dan kode	Kebijakan keamanan Anda berlaku untuk semua aplikasi. Tentukan sekali dan AWS Security Agent menerapkannya di mana saja.
GitHub integrasi	GitHub Organisasi terdaftar atau akun pengguna yang berwenang untuk terhubung dengan AWS Security Agent	Daftarkan GitHub organisasi Anda sekali, lalu hubungkan repositori tertentu ke Ruang Agen mana pun sesuai kebutuhan.
Konfigurasi Pusat Identitas IAM	Pengaturan SSO yang mengontrol cara pengguna mengakses AWS Security Agent	Manajemen identitas terpusat berlaku di semua Ruang Agen di organisasi Anda.

> **Important**
>
> Perubahan pada persyaratan keamanan memengaruhi semua tinjauan desain dan tinjauan kode future di semua Ruang Agen. Ulasan yang ada tidak terpengaruh.

Apa yang dicakup per Ruang Agen

Setiap Ruang Agen mewakili aplikasi atau proyek berbeda yang ingin Anda amankan. Sumber daya di tingkat Ruang Agen dicakup untuk aplikasi spesifik itu, memungkinkan tim yang berbeda untuk bekerja secara independen dengan konfigurasi dan penilaian mereka sendiri.

Sumber daya	Apa itu	Mengapa itu dicakup per aplikasi
Konfigurasi uji penetrasi	Menguji konfigurasi untuk fitur tertentu, titik akhir API, atau fungsionalitas dalam aplikasi Anda	Setiap aplikasi memiliki target unik, metode otentikasi, dan batasan ruang lingkup khusus untuk aplikasi itu.
Ulasan desain	Penilaian keamanan arsitektur individu dari dokumen desain	Setiap aplikasi memiliki arsitektur dan dokumen desain sendiri yang dinilai secara independen.
Model ancaman	Penilaian pemodelan ancaman yang membangun ikhtisar sistem dan mengidentifikasi ancaman dari kode sumber, dokumen desain, atau keduanya	Setiap aplikasi memiliki kode dan desain sendiri, dan ancaman dimodelkan secara independen. Model ancaman adalah konfigurasi yang dapat digunakan kembali yang dapat Anda jalankan kembali saat kode dan desain Anda berkembang.
Integrasi	Penyedia sumber dan dokumentasi (GitHub, GitLab, Bitbucket, Server GitHub Perusahaan, dan Confluence) terhubung ke Ruang Agen ini	Aplikasi yang berbeda bergantung pada sumber dan dokumentasi yang berbeda. Menghubungkan mereka di tingkat Ruang Agen membuat batas aplikasi tetap jelas.

Sumber daya	Apa itu	Mengapa itu dicakup per aplikasi
Pengaturan peninjauan kode	Konfigurasi kemampuan peninjauan kode termasuk sumber yang terhubung, pengaturan pemindaian, dan pemberdayaan komentar PR	Setiap aplikasi memiliki repositori sendiri dan kebutuhan tinjauan keamanan yang dikonfigurasi secara independen.
Pengaturan remediasi uji penetrasi	Konfigurasi repositori yang terhubung dapat menerima permintaan tarik perbaikan otomatis untuk temuan pengujian penetrasi	Tim mengontrol tempat AWS Security Agent dapat mengirimkan perubahan kode berdasarkan alur kerja aplikasi mereka.
Tugas pengguna	Pengguna yang memiliki akses ke Ruang Agen khusus ini	Tim hanya melihat penilaian keamanan untuk aplikasi yang menjadi tanggung jawab mereka, menjaga pekerjaan tetap teratur dan fokus.

> **Tip**
>
> Sebaiknya buat satu Ruang Agen per aplikasi atau proyek untuk menjaga batasan yang jelas antar tim dan mengatur penilaian keamanan secara efektif.

Bagaimana GitHub repositori masuk ke dalam hierarki

GitHub repositori terintegrasi melalui proses multi-langkah yang menghubungkan sumber daya organisasi ke aplikasi tertentu:

1. Mendaftar di tingkat penyewa - Otorisasi GitHub Aplikasi AWS Security Agent untuk GitHub organisasi atau akun pengguna Anda satu kali
2. Connect di level Agent Space - Pilih repositori tertentu untuk terhubung ke setiap Ruang Agen
3. Konfigurasikan penggunaan per repositori - Aktifkan kemampuan khusus untuk setiap repositori yang terhubung:
   - Ulasan kode - Pemindaian kode sumber lengkap dan analisis permintaan tarik otomatis
   - Konteks pengujian penetrasi - Pemahaman aplikasi dari kode sumber selama tes penetrasi

- Remediasi kode otomatis - Permintaan tarik otomatis dengan perbaikan kerentanan untuk tinjauan kode dan temuan pengujian penetrasi

Sebuah repositori tunggal dapat dihubungkan ke beberapa Ruang Agen dengan kemampuan berbeda yang diaktifkan di masing-masing.

Perbedaan utama antara kemampuan keamanan

Setiap kemampuan keamanan di AWS Security Agent mengikuti model alur kerja yang berbeda berdasarkan cara tim keamanan menggunakannya.

Pengujian penetrasi: Konfigurasi yang dapat digunakan kembali dengan eksekusi independen

Tes penetrasi menggunakan model konfigurasi-dan-jalankan yang mendukung pengujian keamanan berulang:

- Buat sekali, jalankan berkali-kali - Tentukan konfigurasi untuk target tertentu (titik akhir API, area fitur) dengan batas cakupan, otentikasi, dan parameter pengujian
- Eksekusi independen - Jalankan konfigurasi yang sama beberapa kali saat Anda meningkatkan keamanan. Setiap eksekusi independen dan menghasilkan temuan baru

Model ini mendukung validasi keamanan berkelanjutan saat Anda mengembangkan dan menerapkan peningkatan.

Ulasan desain: One-off penilaian dengan kloning

Tinjauan desain adalah penilaian independen yang tidak mengikuti model konfigurasi yang dapat digunakan kembali:

- Penilaian tunggal - Setiap tinjauan desain menganalisis dokumen yang diunggah satu kali terhadap persyaratan keamanan organisasi Anda
- Tidak dapat dijalankan kembali - Ulasan desain tidak dapat digunakan kembali. Anda tidak dapat menjalankan kembali ulasan yang sama
- Kloning untuk pembaruan - Kloning tinjauan desain yang ada untuk membuat ulasan baru dengan dokumen asli yang dimuat sebelumnya, memungkinkan Anda memperbarui dokumen dan menjalankan analisis baru

Model ini mendukung penilaian keamanan arsitektur point-in-time.

Ulasan kode: Konfigurasi yang dapat digunakan kembali dengan pemindaian sesuai permintaan dan analisis PR otomatis

Ulasan kode menyediakan dua mode operasi untuk mengamankan kode sumber Anda:

- Ulasan kode lengkap (aplikasi web) - Buat konfigurasi tinjauan kode yang memilih GitHub repositori atau sumber S3, lalu jalankan pemindaian komprehensif sesuai permintaan. Setiap proses melakukan analisis statis di seluruh kode sumber lengkap Anda dan menghasilkan temuan dengan panduan remediasi. Anda dapat menjalankan kembali konfigurasi peninjauan kode yang sama saat kode Anda berkembang.
- Tarik komentar permintaan (GitHub) - Aktifkan analisis otomatis untuk GitHub repositori yang terhubung. AWS Security Agent secara otomatis meninjau permintaan tarik ketika mereka ditandai sebagai siap untuk ditinjau dan memposting temuan keamanan sebagai komentar langsung di GitHub.

Kedua mode menggunakan pengaturan peninjauan kode yang dikonfigurasi (kerentanan keamanan, persyaratan khusus, atau keduanya) dan mendukung perbaikan kode otomatis melalui permintaan tarik.

Model ancaman: Konfigurasi yang dapat digunakan kembali dengan proses sesuai permintaan

Model ancaman menggunakan model konfigurasi-dan-jalankan yang mendukung penilaian berulang arsitektur Anda:

- Buat sekali, jalankan berkali-kali — Tentukan model ancaman dengan memilih kode sumber sebagai sumber, mengunggah dokumen desain sebagai dokumen lingkup, atau keduanya. Jalankan sesuai permintaan dan jalankan kembali saat kode dan desain Anda berkembang.
- Input fleksibel — Jalankan model ancaman pada kode sumber saja, desain dokumen saja, atau keduanya. Dokumen lingkup menentukan apa yang agen fokuskan analisisnya; kode sumber menyediakan konteks tentang sistem Anda yang ada.
- Gambaran umum dan ancaman sistem — Setiap proses menghasilkan ikhtisar sistem yang menjelaskan arsitektur aplikasi Anda, batas kepercayaan, aliran data, dan postur keamanan, bersama dengan serangkaian ancaman yang diklasifikasikan menurut kategori STRIDE dengan tingkat keparahan, bukti, dan rekomendasi yang dapat ditindaklanjuti.

Memahami hubungan sumber daya

Hirarki menentukan tempat Anda mengonfigurasi dan mengakses sumber daya yang berbeda:

Di AWS Management Console:

- Konfigurasikan sumber daya tingkat penyewa (persyaratan keamanan, GitHub integrasi, Pusat Identitas IAM)
- Buat dan kelola Ruang Agen
- Konfigurasikan pengaturan Ruang Agen (repositori terhubung, pemberdayaan peninjauan kode, remediasi uji penetrasi)

Dalam Aplikasi Web Agen Keamanan:

- Membuat dan mengelola konfigurasi pengujian penetrasi dan eksekusi uji
- Membuat dan mengelola ulasan desain
- Membuat, mengelola, dan menjalankan ulasan kode terhadap repositori yang terhubung dan sumber S3
- Membuat, mengelola, dan menjalankan model ancaman terhadap kode sumber, dokumen cakupan, atau keduanya
- Lihat temuan dari tes penetrasi, ulasan kode, ulasan desain, dan model ancaman

Dalam GitHub:

- Lihat temuan tinjauan kode permintaan tarik sebagai komentar permintaan tarik
- Menerima permintaan tarik remediasi otomatis untuk tinjauan kode dan temuan pengujian penetrasi (saat diaktifkan di Ruang Agen)

> **Note**
>
> Temuan tinjauan kode permintaan tarik muncul di GitHub. Tinjauan kode lengkap, uji penetrasi, dan temuan tinjauan desain muncul di Aplikasi Web Agen Keamanan.

Memulai

Daftar, temukan titik awal yang tepat untuk tujuan Anda, dan jalankan penilaian pertama Anda dengan memulai cepat.

Mendaftar untuk AWS akun

Untuk memulai AWS, Anda memerlukan AWS akun. Untuk informasi tentang membuat AWS akun, lihat Memulai AWS akun di Panduan Referensi Manajemen AWS Akun.

Pilih jalan Anda

Temukan titik awal yang tepat untuk apa yang ingin Anda lakukan dengan AWS Security Agent. Gunakan tabel di bawah ini untuk mencocokkan tujuan Anda dengan tugas, lalu langsung ke halaman itu. Jika Anda baru mengenal AWS Security Agent, baca the section called “Apa itu Agen Keamanan AWS?” terlebih dahulu untuk ikhtisar singkat.

Aku ingin...	Siapa	Mulai di sini
Pahami apa yang dilakukan AWS Security Agent sebelum mengaturnya	Semua orang	the section called “Apa itu Agen Keamanan AWS?”
Siapkan layanan dan buat ruang agen	Admin	the section called “Siapkan Agen Keamanan AWS”
Uji aplikasi langsung atau penerapan saya untuk kerentanan yang dapat dieksploitasi	Pengguna	the section called “Quickstart: Jalankan tes penetrasi”
Pindai kode sumber saya untuk mengetahui kerentanan dan pelanggaran kebijakan	Pengguna	the section called “Mulai cepat: Jalankan tinjauan kode”
Model bagaimana aplikasi saya dapat diserang (STRIDE)	Pengguna	the section called “Mulai cepat: Jalankan model ancaman”

Aku ingin...	Siapa	Mulai di sini
Dapatkan umpan balik keamanan pada desain sebelum saya menulis kode	Pengguna	the section called “Buat tinjauan desain”
Pindai kode tanpa meninggalkan IDE saya (Kiro atau Claude Code)	Developer	the section called “Jalankan pemindaian keamanan kode dari IDE Anda”
Pindai hanya baris saya yang berubah sebelum menggabungkan	Developer	the section called “Jalankan pemindaian kode diferensial dengan S3”
Dapatkan komentar keamanan otomatis pada permintaan tarik dan gabungkan permintaan	Admin	the section called “Aktifkan peninjauan kode”
Tinjau temuan dan putuskan apa yang harus diperbaiki terlebih dahulu	Pengguna	Tes penetrasi, Tinjauan kode, Model ancaman, Tinjauan desain

> **Note**
>
> AWS Security Agent menggunakan tiga peran, yang dapat Anda identifikasi berdasarkan antarmuka tempat Anda bekerja: Admin bekerja di AWS Management Console (penyiapan dan konfigurasi), Pengguna bekerja di aplikasi web (menjalankan penilaian dan meninjau temuan), dan Developer bekerja di GitHub atau IDE seperti Kiro atau Claude Code. Satu orang dapat memegang lebih dari satu peran. Untuk definisi lengkap, lihatthe section called “Cara kerja AWS Security Agent”.

Quickstart: Jalankan tes penetrasi

Mulai cepat ini memandu Anda menjalankan pengujian penetrasi pertama (pentest) dengan AWS Security Agent. Tes penetrasi menjalankan aplikasi yang Anda gunakan terhadap domain target yang diverifikasi dan mengembalikan temuan keamanan, masing-masing dengan peringkat keparahan dan bukti pendukung. Ini mencakup aplikasi yang dapat diakses publik; untuk menguji satu yang dihosting di VPC pribadi, lihat. the section called “Aktifkan uji penetrasi”

> **Note**
>
> Anda memerlukan akses ke AWS Management Console untuk menyiapkan AWS Security Agent dan menentukan cakupan pengujian, serta akses ke aplikasi web untuk membuat dan menjalankan pengujian penetrasi.

Prasyarat

Sebelum memulai, pastikan Anda memiliki:

- Akses ke AWS Management Console dengan izin untuk menyiapkan AWS Security Agent.
- Domain target langsung yang menghosting aplikasi yang ingin Anda uji.
- Kemampuan untuk menambahkan catatan DNS untuk domain tersebut, atau zona yang dihosting Route 53 di akun AWS yang sama, sehingga Anda dapat memverifikasi kepemilikan.
- (Opsional) Sebuah repositori kode sumber (GitHub, GitLab, atau Bitbucket) untuk aplikasi Anda, untuk memberikan agen lebih banyak konteks.

Langkah 1: Siapkan AWS Security Agent di konsol AWS

Jika Anda belum menyiapkan AWS Security Agent, selesaikan penyiapan awal:

1. Arahkan ke AWS Security Agent di AWS Management Console.
2. Pilih Siapkan AWS Security Agent.
3. Buat Ruang Agen. Ruang agen dapat digunakan oleh banyak pengguna dan harus spesifik untuk setiap aplikasi yang ingin Anda uji. Masukkan nama dan deskripsi. Nama harus mengidentifikasi aplikasi yang ingin Anda uji penetrasi.
4. Pilih IAM-only akses di bawah Konfigurasi akses pengguna.
   - Quickstart ini tidak mencakup mengaktifkan single sign-on (SSO) dengan IAM Identity Center, yang memungkinkan pengguna mengakses aplikasi web langsung dari AWS Console.
   - Untuk memungkinkan pengguna tanpa AWS Management Console mengakses pengujian penetrasi, aktifkan integrasi IAM Identity Center. Lihat perinciannya di the section called “Berikan pengguna akses ke aplikasi web AWS Security Agent”.
5. Pilih Siapkan AWS Security Agent.

> **Note**
>
> Saat Anda memilih Pengaturan, AWS Security Agent membuat Ruang Agen Anda dan membuat aplikasi web tempat pengguna dapat menjalankan tes penetrasi, ulasan kode, model ancaman, dan ulasan desain.

Langkah 2: Aktifkan pengujian penetrasi dan verifikasi domain Anda

> **Note**
>
> Di konsol AWS Anda menentukan ruang lingkup apa yang dapat diuji. Pengguna kemudian menjalankan tes penetrasi tertentu dalam lingkup itu dari aplikasi web.

1. Dari bilah sisi kiri, pilih Ruang Agen, lalu pilih Ruang Agen yang Anda buat di Langkah 1.
2. Pilih tab Tes Penetrasi, lalu pilih Siapkan tes penetrasi untuk membuka wizard.
3. Konfigurasikan domain - Masukkan domain target yang ingin Anda uji dan pilih metode verifikasi, DNS TXT Record atau HTTP Route. Domain harus live dan host aplikasi yang ingin Anda uji. Pilih Berikutnya.
4. Verifikasi domain - Verifikasi kepemilikan setiap domain dalam tabel domain Target. AWS Security Agent menjalankan pengujian hanya terhadap domain terverifikasi. Untuk langkah-langkah terperinci dan nilai yang tepat untuk disalin, lihatthe section called “Aktifkan domain aplikasi untuk pengujian penetrasi”.
   - Route 53 domain di akun AWS yang sama - Pilih domain dan pilih One-click verifikasi. AWS Security Agent membuat catatan DNS dan menyelesaikan verifikasi untuk Anda.
   - Catatan DNS TXT (penyedia DNS lainnya) - Di tabel domain Target, salin nama Rekam dan nilai Rekam, tambahkan sebagai catatan TXT dengan penyedia DNS Anda, lalu pilih domain dan pilih Verifikasi. Perubahan DNS membutuhkan waktu untuk menyebar, jadi verifikasi mungkin tidak segera selesai.
   - Rute HTTP - Buat file `.well-known/aws/securityagent-domain-verification.json` di server web Anda, tambahkan token verifikasi dalam format`{"tokens": ["<token>"]}`, lalu pilih domain dan pilih Verifikasi.
5. (Opsional) Konfigurasikan kemampuan tambahan - Tambahkan sumber daya opsional seperti grup CloudWatch log dan kredensyal. Bagian akses Layanan sudah dikonfigurasi sebelumnya: AWS

Security Agent membuat peran layanan dengan izin yang diperlukan kecuali Anda memilih peran IAM yang ada.

Langkah 3: Hubungkan kode sumber (opsional)

Menyambungkan penyedia kode sumber memberikan konteks AWS Security Agent tentang aplikasi Anda dan meningkatkan cakupan pengujian penetrasi. Langkah ini bersifat opsional.

1. Pada tab Penetration test, pilih Add on the Connect a source code provider for penetration testing banner di bagian atas halaman.
2. Daftarkan dan hubungkan penyedia Anda, lalu pilih repositori yang akan disediakan untuk pengujian penetrasi.

Untuk alur integrasi lengkap, lihat Connect AWS Security Agent ke GitHub repositori atau topik connect untuk penyedia Anda.

Langkah 4: Buat dan jalankan tes penetrasi

> **Note**
>
> Anda membuat dan menjalankan pengujian penetrasi di aplikasi web AWS Security Agent. Kami merekomendasikan menjalankan pengujian terhadap lingkungan pengujian yang sangat mencerminkan produksi.

1. Luncurkan aplikasi web: pilih tab Aplikasi Web, lalu akses Admin.
2. Di sidebar kiri, pilih Tes penetrasi, lalu Buat tes penetrasi.
3. Detail tes penetrasi - Siapkan ruang lingkup pengujian dan sumber log:
   a. Masukkan nama Pentest.
   b. Di bawah URL Target, pilih atau masukkan domain terverifikasi untuk diuji (misalnya,`https://example.com`). Hanya domain terverifikasi yang dapat diuji, dan sub-domain dicakup secara otomatis.
   c. (Opsional) Di bawah URL yang Dapat Diakses, tambahkan domain yang harus dijangkau agen untuk login dan navigasi tetapi tidak boleh menyerang, seperti penyedia identitas, CDN, atau API di luar domain target Anda. AWS Security Agent dapat menjangkau domain ini tetapi tidak mengujinya; hanya domain target Anda yang diserang.

d. Tinjau aturan konfigurasi Jaringan untuk mengonfirmasi titik akhir mana yang dapat dan tidak dapat menerima lalu lintas selama pengujian.

e. Di bawah Izin, pilih peran Layanan dan, secara opsional, grup CloudWatch log. Pilih Berikutnya.

4. (Opsional) Sumber Daya VPC - Konfigurasikan VPC jika target Anda berada di jaringan pribadi yang tidak dapat dijangkau oleh publik. Lihat the section called “Aktifkan uji penetrasi”.
5. (Opsional) Sumber Daya Otentikasi - Berikan kredensyal sehingga agen dapat mencapai jalur non-publik yang diautentikasi. Sebaiknya tambahkan ini untuk memperluas cakupan dan kerentanan permukaan di balik login.
6. (Opsional) Konfigurasi tambahan - Tambahkan konteks aplikasi seperti file, GitHub repositori, atau sumber S3 untuk meningkatkan kualitas pencarian. Anda juga dapat mengaktifkan remediasi kode otomatis dan mengatur opsi run lainnya di sini.
7. Pilih Buat dan jalankan untuk memulai pengujian sekarang, atau Buat pentest untuk menyimpannya dan menjalankannya nanti.

Langkah 5: Tinjau temuan uji penetrasi

1. Tes penetrasi dapat memakan waktu beberapa jam untuk diselesaikan. Paling lengkap dalam waktu 16 jam, tergantung pada ukuran dan kompleksitas aplikasi Anda.
2. Proses dimulai dengan fase Preflight yang memvalidasi penyiapan sebelum pengujian dimulai: ini menyiapkan infrastruktur logging, memeriksa apakah titik akhir target Anda dapat dijangkau, dan menyiapkan lingkungan pengujian. Jika pemeriksaan preflight gagal (misalnya, domain target yang tidak dapat diselesaikan), proses akan berhenti sebelum pengujian dimulai. Buka tab Preflight untuk melihat pemeriksaan mana yang gagal, selesaikan, dan mulai proses baru.
3. Saat proses selesai, buka dan tinjau tab ikhtisar Run, Log, dan Temuan.
4. Pada tab Temuan, pilih temuan untuk melihat deskripsi, tingkat keparahan, jenis risiko, dan bukti pendukungnya.

Untuk lebih jelasnya, lihat the section called “Buat tes penetrasi” danthe section called “Tinjau temuan dari tes penetrasi”.

Mulai cepat: Jalankan tinjauan kode

Mulai cepat ini memandu Anda menjalankan peninjauan kode pertama Anda dengan AWS Security Agent. AWS Security Agent memindai repositori kode sumber Anda untuk mengetahui kerentanan keamanan dan kepatuhan terhadap persyaratan keamanan organisasi Anda.

> **Note**
>
> Anda memerlukan akses ke AWS Management Console untuk menyiapkan AWS Security Agent, dan akses ke aplikasi web untuk membuat dan menjalankan ulasan kode.

Prasyarat

Sebelum memulai, pastikan Anda memiliki:

- Akses ke AWS Management Console dengan izin untuk menyiapkan AWS Security Agent.
- Repositori kode sumber (GitHub, GitLab, atau Bitbucket) atau sumber Amazon S3 yang berisi kode yang ingin Anda tinjau.

Langkah 1: Siapkan AWS Security Agent di konsol AWS

Jika Anda belum menyiapkan AWS Security Agent, selesaikan penyiapan awal:

1. Arahkan ke AWS Security Agent di AWS Management Console.
2. Pilih Siapkan Agen Keamanan AWS.
3. Buat Ruang Agen. Ruang agen dapat digunakan oleh banyak pengguna dan harus spesifik untuk setiap aplikasi yang ingin Anda uji. Masukkan nama dan deskripsi untuk ruang agen pertama Anda. Nama ini muncul untuk pengguna di aplikasi web. Nama harus mengidentifikasi aplikasi yang kodenya ingin Anda tinjau.
4. Pilih IAM-only akses di bawah Konfigurasi akses pengguna.
   - Quickstart ini tidak mencakup mengaktifkan single sign-on (SSO) dengan IAM Identity Center. Hal ini memungkinkan pengguna untuk langsung mengakses aplikasi web AWS Security Agent, dari AWS Console.

- Agar pengguna tanpa AWS Management Console mengakses ulasan kode, aktifkan integrasi IAM Identity Center. Lihat perinciannya di the section called “Berikan pengguna akses ke aplikasi web AWS Security Agent”.

5. Pilih Siapkan Agen Keamanan AWS.

> **Note**
>
> Saat Anda memilih Pengaturan, AWS Security Agent membuat Ruang Agen Anda dan membuat aplikasi web tempat pengguna dapat menjalankan tes penetrasi, ulasan kode, model ancaman, dan ulasan desain.

Langkah 2: Aktifkan dan konfigurasikan tinjauan kode

> **Note**
>
> Jika Anda sudah memiliki GitHub repositori atau bucket S3 yang terhubung ke Ruang Agen Anda (misalnya, melalui pengaturan pengujian penetrasi), peninjauan kode sudah diaktifkan. Anda dapat melewati langkah ini dan langsung menuju ke aplikasi web.

Buka wizard penyiapan peninjauan kode

1. Dari bilah sisi kiri, pilih Ruang Agen dan kemudian pilih Ruang Agen Anda.
2. Pilih Aktifkan tinjauan kode dari header atau tab Ulasan kode.

Connect integrasi, repositori, dan bucket

1. (Jika Anda belum memiliki GitHub integrasi) Buat GitHub pendaftaran. Jika Anda sudah memilikinya, lompat ke langkah berikutnya.
   a. Di bagian Integrasi terhubung, pilih Tambah dan kemudian Buat pendaftaran baru.
   b. Pilih GitHubdan pilih Berikutnya.
   c. Pilih Instal dan otorisasi, lalu selesaikan instalasi di GitHub:
      i. Pilih GitHub pengguna atau organisasi yang memiliki repositori yang ingin Anda tinjau.
      ii. Pilih Semua repositori atau Hanya pilih repositori.

iii. Pilih Instal & Otorisasi dan selesaikan GitHub otentikasi.

d. Kembali ke AWS Management Console, masukkan nama Pendaftaran dan konfirmasikan jenis Akun cocok dengan tempat Anda menginstal GitHub Aplikasi.

e. Pilih Connect untuk menyimpan pendaftaran.

Untuk alur GitHub integrasi lengkap, lihat Connect AWS Security Agent ke GitHub repositori.

2. Connect GitHub repositori. Di bagian Integrasi terhubung, pilih Tambah, lalu pilih GitHub pendaftaran Anda. GitHubWizard Connect dua langkah terbuka:

a. Pada Connect GitHub repositori, pilih repositori yang akan disertakan dan pilih Next.

b. Pada kemampuan Kelola, alihkan yang berikut ini per repositori:

- Komentar tinjauan kode - Biarkan AWS Security Agent memposting temuan keamanan sebagai komentar pada permintaan tarik di repositori.
- Remediasi otomatis — Memungkinkan pengguna aplikasi web AWS Security Agent meminta permintaan tarik yang memperbaiki temuan.

c. Pilih Simpan untuk kembali ke wizard pengaturan.

3. (Opsional) Hubungkan sumber S3. Di bagian bucket S3, pilih Tambahkan sumber daya S3 dan masukkan URI S3 untuk bucket yang berisi kode sumber, atau pilih Browse untuk memilihnya.

4. Pilih pengaturan peninjauan Kode Anda. Default, Persyaratan keamanan dan temuan kerentanan, menganalisis kode untuk kepatuhan terhadap persyaratan keamanan yang telah Anda aktifkan dan kerentanan umum.

5. Pilih Berikutnya.

Konfigurasi opsional

1. Konfigurasikan grup CloudWatch log opsional dan akses layanan. Peran layanan default sudah dikonfigurasi sebelumnya dengan izin yang diperlukan.

2. Pilih Simpan.

Langkah 3: Buat dan jalankan tinjauan kode

> **Note**
>
> Anda membuat dan menjalankan ulasan kode hanya di aplikasi web AWS Security Agent.

1. Pilih tab Aplikasi Web dan kemudian akses Admin untuk meluncurkan aplikasi web AWS Security Agent.
2. Di bilah sisi kiri, pilih Ulasan kode.
3. Pilih Buat ulasan kode.
4. Konfigurasikan tinjauan kode:
   a. Masukkan Judul yang mengidentifikasi cakupan ulasan ini (misalnya, “billing-service-security-review”).
   b. Di bawah Sumber, pilih GitHub repositori yang Anda sambungkan ke Ruang Agen di Langkah 2, atau masukkan sumber S3 yang ingin Anda pindai.
   c. Pilih peran Layanan dari peran yang dikonfigurasi.
   d. (Opsional) Pilih Aktifkan remediasi kode otomatis agar AWS Security Agent secara otomatis mengirimkan permintaan tarik dengan perbaikan untuk semua temuan.
5. Pilih Buat ulasan kode.
6. Pada halaman detail peninjauan kode, pilih Mulai ulasan.

Langkah 4: Tinjau temuan tinjauan kode

1. Peninjauan kode biasanya memakan waktu 30-60 menit tergantung pada ukuran basis kode Anda.
2. Proses dimulai dengan fase Preflight yang memvalidasi penyiapan sebelum pemindaian dimulai: ini mengonfirmasi AWS Security Agent dapat menarik kode sumber Anda dari repositori atau sumber S3 Anda dan menyiapkan lingkungan pemindaian. Jika pemeriksaan preflight gagal (misalnya, pemeriksaan tidak dapat mengakses sumber Anda), proses akan berhenti sebelum analisis statis. Buka tab Preflight untuk melihat pemeriksaan mana yang gagal, selesaikan, dan mulai proses baru.
3. Setelah selesai, navigasikan ke proses selesai dan pilih tab Temuan.
4. Tinjau temuan dalam tampilan detail-daftar:
   a. Pilih temuan dari panel kiri untuk melihat detailnya.
   b. Tinjau bagian Deskripsi, Lokasi Kode, Bukti, dan Perbaikan yang Disarankan.
   c. Gunakan kode Remediate untuk menghasilkan permintaan tarik dengan perbaikan, atau tinjau PR remediasi otomatis jika Anda mengaktifkan opsi itu.

Untuk lebih jelasnya, lihat the section called “Buat ulasan kode” danthe section called “Tinjau temuan dari tinjauan kode”.

Mulai cepat: Jalankan model ancaman

Mulai cepat ini memandu Anda menjalankan model ancaman pertama Anda dengan AWS Security Agent. Model ancaman menganalisis arsitektur aplikasi Anda dan menghasilkan ikhtisar sistem (cara AWS Security Agent memahami sistem Anda) dan serangkaian ancaman (bagaimana hal itu dapat diserang, masing-masing dengan tingkat keparahan, klasifikasi STRIDE, dan rekomendasi). Anda dapat menjalankan model ancaman pada dokumen desain (dokumen lingkup) untuk menentukan fokus, kode sumber (sumber) untuk memberikan konteks tentang sistem yang ada, atau keduanya.

> **Note**
>
> Anda memerlukan akses ke AWS Management Console untuk menyiapkan AWS Security Agent, dan akses ke aplikasi web untuk membuat dan menjalankan model ancaman.

Langkah 1: Siapkan AWS Security Agent di konsol AWS

Jika Anda belum menyiapkan AWS Security Agent, selesaikan penyiapan awal:

1. Arahkan ke AWS Security Agent di AWS Management Console.
2. Pilih Siapkan AWS Security Agent.
3. Buat Ruang Agen. Ruang agen dapat digunakan oleh banyak pengguna dan harus spesifik untuk setiap aplikasi yang ingin Anda amankan. Masukkan nama dan deskripsi untuk ruang agen pertama Anda. Nama harus mengidentifikasi aplikasi yang ingin Anda ancaman model.
4. Pilih IAM-only akses di bawah Konfigurasi akses pengguna.
   - Quickstart ini tidak mencakup mengaktifkan single sign-on (SSO) dengan IAM Identity Center. Hal ini memungkinkan pengguna untuk langsung mengakses aplikasi web AWS Security Agent dari AWS Console.
   - Jika Anda ingin mengaktifkan pengguna tanpa akses AWS Management Console untuk melakukan tugas seperti memulai model ancaman, Anda harus mengaktifkan integrasi Pusat Identitas IAM.
5. Pilih Siapkan AWS Security Agent.

> **Note**
>
> Saat Anda memilih Pengaturan, AWS Security Agent membuat Ruang Agen Anda dan membuat aplikasi web tempat pengguna dapat menjalankan tes penetrasi, ulasan kode, model ancaman, dan ulasan desain.

Langkah 2: Hubungkan kode sumber

> **Note**
>
> Jika Anda sudah memiliki repositori atau bucket S3 yang terhubung ke Ruang Agen Anda (misalnya, melalui peninjauan kode atau pengaturan pengujian penetrasi), Anda dapat melewati langkah ini dan langsung menuju ke aplikasi web. Anda selalu dapat menjalankan model ancaman pada dokumen cakupan yang diunggah tanpa menghubungkan kode sumber apa pun.

Connect repositori atau bucket S3 yang berisi kode sumber yang ingin digunakan agen sebagai konteks untuk memahami sistem yang ada:

1. Dari bilah sisi kiri, pilih Ruang Agen dan kemudian pilih Ruang Agen Anda.
2. Arahkan ke bagian Integrasi untuk menghubungkan penyedia kode sumber Anda.
3. Atau, hubungkan bucket S3 yang berisi kode sumber Anda.

Untuk alur integrasi lengkap, lihat Connect AWS Security Agent ke GitHub repositori.

Langkah 3: Buat dan jalankan model ancaman

> **Note**
>
> Anda membuat dan menjalankan model ancaman di aplikasi web AWS Security Agent.

1. Luncurkan aplikasi web dari tab Aplikasi Web di AWS Management Console. Atau, jika Anda memiliki Pusat Identitas IAM yang dikonfigurasi, masuk langsung.

2. Di bilah sisi kiri, pilih model Ancaman.
3. Pilih Buat model ancaman.
4. Konfigurasikan model ancaman:
   a. Masukkan Judul yang mengidentifikasi ruang lingkup model ancaman ini (misalnya, “layanan penagihan” atau “checkout-api”).
   b. Berikan setidaknya satu masukan:
      - Di bawah dokumen Scope, unggah dokumen desain (DOC, DOCX, JPEG, MD, PDF, PNG, atau TXT), masukkan URI S3, atau pilih halaman Confluence.
      - Di bawah Sumber, pilih repositori dari integrasi Anda yang terhubung atau masukkan URI S3 yang berisi kode sumber.

      > **Tip**
      >
      > Sediakan dokumen sumber dan ruang lingkup untuk mencakup model ancaman ke desain tertentu (misalnya, dokumen desain fitur) sambil memberi agen kode sumber Anda sebagai konteks untuk memahami sistem Anda yang ada.

   c. Pilih peran Layanan dari peran yang dikonfigurasi.
   d. (Opsional) Pilih grup Log untuk CloudWatch log.
5. Pilih Buat model ancaman.
6. Pada halaman detail model ancaman, pilih Mulai jalankan.

Langkah 4: Tinjau ancaman

1. Lari berlangsung melalui tugas analisisnya. Anda dapat memantau kemajuan pada tab Preflight dan melihat detail tugas di tab Log.
2. Setelah selesai, status run berubah menjadi Selesai.
3. Pilih tab Ikhtisar untuk meninjau ikhtisar sistem dan distribusi tingkat keparahan.
4. Pilih tab Ancaman untuk meninjau ancaman yang diidentifikasi:
   a. Pilih ancaman dari daftar untuk melihat detailnya di panel samping.
   b. Tinjau Pernyataan, Keparahan, kategori STRIDE, Rekomendasi, Bukti, dan bidang lainnya.
   c. Perbarui status ancaman menjadi Diselesaikan atau Dihentikan saat Anda mengatasi atau melakukan triase setiap ancaman.

Untuk lebih jelasnya, lihat the section called “Buat model ancaman” danthe section called “Meninjau ancaman dari model ancaman”.

Untuk administrator (konsol)

Sebagai administrator, Anda menyiapkan AWS Security Agent di AWS Management Console dan mengonfigurasi Ruang Agen yang diakses pengguna melalui aplikasi web AWS Security Agent. Setiap Ruang Agen mewakili lingkungan yang berbeda dengan izin dan sumber daya tertentu.

Sebaiknya buat Ruang Agen unik untuk setiap aplikasi yang ingin Anda uji. Misalnya, jika Anda memiliki dua proyek internal—aplikasi penagihan dan aplikasi pelacakan tugas—Anda harus membuat dua Ruang Agen terpisah.

Contoh konfigurasi

Pertimbangkan administrator yang menyiapkan Ruang Agen untuk menilai keamanan aplikasi penagihan internal. Administrator akan:

- Verifikasi domain (sepertibeta.billing.example.com)
- Connect ke GitHub dan aktifkan peninjauan kode
- Konfigurasikan akses jaringan dengan menetapkan VPC, subnet, dan grup keamanan yang sesuai untuk pengujian penetrasi

Saat pengguna memulai uji penetrasi atau tinjauan desain, mereka dapat memilih dari sumber daya yang telah dikonfigurasi sebelumnya ini, bekerja di dalam pagar pembatas yang telah Anda tetapkan sambil mempertahankan fleksibilitas untuk kebutuhan penilaian spesifik mereka.

Siapkan dan buat Ruang Agen

Siapkan AWS Security Agent untuk organisasi Anda dan buat Ruang Agen yang mencakup setiap sumber daya dan penilaian aplikasi.

Siapkan Agen Keamanan AWS

Konfigurasikan AWS Security Agent untuk organisasi Anda dengan membuat Ruang Agen pertama Anda dan menetapkan cara pengguna mengakses aplikasi web.

Penyiapan awal ini memungkinkan administrator untuk mengonfigurasi kemampuan keamanan di AWS Console dan memberi pengguna akses ke tinjauan desain dan pengujian penetrasi melalui

Aplikasi Web Agen Keamanan. Setelah penyiapan satu kali ini, Anda dapat membuat Ruang Agen tambahan dengan proses yang disederhanakan.

Gambaran umum

Memahami Ruang Agen

Ruang Agen adalah ruang kerja khusus untuk mengamankan aplikasi atau proyek tertentu. Ini berisi semua ulasan keamanan, GitHub repositori yang terhubung secara opsional, konfigurasi uji penetrasi, hasil, dan temuan untuk aplikasi itu.

Agen Spaces membantu Anda mengatur pekerjaan keamanan dengan menjaga penilaian keamanan setiap aplikasi terpisah, memungkinkan tim untuk fokus pada aplikasi spesifik mereka. Sebaiknya buat satu Ruang Agen per aplikasi atau proyek untuk mempertahankan batasan yang jelas.

Persyaratan keamanan didefinisikan di tingkat organisasi dan berlaku di semua Ruang Agen, sementara setiap Ruang Agen mempertahankan dokumen desain sendiri, repositori kode, konfigurasi pengujian penetrasi, hasil pengujian penetrasi, dan temuan keamanan.

Apa yang termasuk dalam Ruang Agen

Setiap Ruang Agen berisi:

- GitHub Repositori yang terhubung secara opsional terkait dengan aplikasi atau proyek
- Ulasan desain sebelumnya untuk aplikasi
- Konfigurasi pengujian penetrasi dan batasan khusus untuk aplikasi
- Hasil tes pengujian penetrasi dan temuan keamanan

Apa yang akan Anda konfigurasikan selama pengaturan

Selama penyiapan pertama kali ini, Anda akan membuat keputusan organisasi yang berlaku untuk semua Ruang Agen:

- Metode akses - Pilih cara pengguna mengakses Aplikasi Web Agen Keamanan (IAM Identity Center SSO atau IAM-only akses melalui AWS Console)
- Izin - Konfigurasikan peran IAM yang digunakan aplikasi web untuk mengakses layanan AWS
- First Agent Space - Buat Ruang Agen awal Anda dengan nama dan deskripsi

> **Note**
>
> Setelah menyelesaikan pengaturan ini, membuat Ruang Agen tambahan lebih sederhana - cukup berikan nama dan deskripsi. Lihat the section called “Buat Ruang Agen” untuk detail tentang membuat Ruang Agen berikutnya.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Izin untuk membuat dan mengelola sumber daya AWS Security Agent
- Memahami persyaratan manajemen identitas organisasi Anda
- (Opsional) Akun AWS dengan izin untuk membuat peran IAM dan mengonfigurasi Pusat Identitas IAM (jika menggunakan akses SSO)
- (Opsional) Peran IAM yang ada jika Anda ingin menggunakan konfigurasi izin khusus

Langkah 1: Buat Ruang Agen pertama Anda

Tentukan properti dasar Ruang Agen pertama Anda yang akan ditampilkan kepada pengguna di aplikasi web.

1. Pada halaman konsol AWS Security Agent, pilih Siapkan Agen Keamanan.
2. Di bidang nama Ruang Agen, masukkan nama untuk Ruang Agen Anda.

> **Note**
>
> Nama Ruang Agen ditampilkan kepada pengguna di aplikasi web dan membantu mengidentifikasi aplikasi atau proyek mana yang diwakili oleh ruang ini.

3. (Opsional) Di bidang Deskripsi, berikan deskripsi yang membantu membedakan tujuan Ruang Agen.

> **Tip**
>
> Deskripsi membantu membedakan tujuan Agen Ruang. Sebaiknya jelaskan aplikasi atau proyek spesifik yang akan diamankan oleh Ruang Agen ini, seperti “Aplikasi web

portal pelanggan” atau “Layanan mikro pemrosesan pembayaran” atau “Platform analitik internal.”

Langkah 2: Pilih metode akses Anda

Pilih bagaimana pengguna akan mengakses Aplikasi Web Agen Keamanan. Anda memilih antara mengaktifkan akses SSO melalui IAM Identity Center atau menyediakan akses melalui AWS Console.

> **Note**
>
> Jika Anda memilih IAM-only akses dan nanti ingin menggunakan IAM Identity Center, Anda harus menghapus penyiapan AWS Security Agent dan menyelesaikan proses penyiapan lagi.

1. Di bagian Metode akses, pilih salah satu opsi berikut:
   - IAM Identity Center (SSO) - Aktifkan akses SSO untuk tim Anda melalui IAM Identity Center. Opsi ini direkomendasikan untuk tim yang membutuhkan manajemen pengguna terpusat dan ingin pengguna mengakses aplikasi web secara langsung tanpa melalui AWS Console.
   - IAM-only akses - Menyediakan akses melalui tautan akses admin di AWS Console. Opsi ini lebih mudah diatur dan cocok untuk tim yang lebih memilih akses berbasis konsol atau tidak memerlukan kemampuan SSO.
2. Jika Anda memilih IAM Identity Center (SSO), lanjutkan ke Langkah 2a di bawah ini.
3. Jika Anda memilih IAM-only akses, lewati ke Langkah 3.

Langkah 2a: Konfigurasikan Pusat Identitas IAM (hanya akses SSO)

Selesaikan langkah-langkah ini hanya jika Anda memilih IAM Identity Center (SSO) sebagai metode akses Anda.

1. Di bagian Connect to IAM Identity Center, tinjau Wilayah AWS.

> **Important**
>
> Aplikasi Anda harus dikonfigurasi di Wilayah yang sama di mana Anda mengaktifkan Pusat Identitas IAM. Wilayah yang ditampilkan adalah tempat Ruang Agen Anda akan dibuat.

Pusat Identitas IAM harus diaktifkan di Wilayah yang sama tempat Anda membuat Ruang Agen.

2. Di bagian Pusat Identitas IAM, pilih salah satu dari berikut ini:
   - Pilih Buat instans akun untuk membuat instans akun Pusat Identitas IAM baru
   - Jika instans organisasi sudah ada di Region yang sama, AWS Security Agent akan secara otomatis terhubung dengannya

> **Note**
>
> Setelah AWS Security Agent terhubung ke IAM Identity Center, Anda dapat mengelola akses dengan menetapkan pengguna di IAM Identity Center ke aplikasi.

3. Jika Anda menghubungkan Active Directory atau penyedia identitas eksternal, tinjau peringatan informasi.

> **Important**
>
> Jika Anda sudah mengelola pengguna di Active Directory atau sumber identitas lain dan berencana untuk menghubungkan sumber identitas tersebut ke IAM Identity Center, batalkan penyiapan dan buka konsol IAM Identity Center terlebih dahulu. Pilih sumber identitas yang ingin Anda sambungkan sebelum menyiapkan AWS Security Agent. Mengubah sumber identitas nantinya dapat menghapus penetapan pengguna yang ada.

Langkah 3: Konfigurasikan izin (Opsional)

Konfigurasikan peran IAM yang digunakan Aplikasi Web Agen Keamanan Anda untuk mengakses layanan, API, dan akun AWS.

1. Temukan konfigurasi Izin - bagian opsional.
2. Jika bagian diciutkan, pilih bagian Konfigurasi izin untuk memperluasnya.
3. Pilih salah satu opsi berikut:
   - Buat peran default - AWS Security Agent secara otomatis membuat peran IAM baru dengan izin yang diperlukan untuk aplikasi web
   - Gunakan peran lain - Pilih peran IAM yang ada dari opsi yang tersedia
4. Tinjau deskripsi peran:

> **Note**
>
> Peran IAM default akan dibuat untuk aplikasi web Anda untuk mengakses layanan, API, dan akun AWS lainnya. Peran akan diberikan izin yang diperlukan untuk operasi penilaian keamanan.

Langkah 4: Pengaturan lengkap

Setelah mengonfigurasi semua pengaturan yang diperlukan, selesaikan pengaturan untuk membuat Ruang Agen pertama Anda dan buat Aplikasi Web Agen Keamanan.

1. Tinjau semua bagian konfigurasi untuk memastikan akurasi.
2. Pilih Siapkan.
3. AWS Security Agent membuat Ruang Agen Anda, membuat aplikasi web, dan mengonfigurasi sumber daya AWS yang diperlukan.

> **Note**
>
> Setelah penyiapan awal, Anda akan memiliki opsi untuk mengonfigurasi kemampuan termasuk batas pengujian penetrasi, persyaratan keamanan, dan GitHub integrasi.

Langkah selanjutnya

Setelah menyiapkan AWS Security Agent:

- Tentukan persyaratan keamanan untuk organisasi Anda
- Konfigurasikan kemampuan pengujian penetrasi termasuk verifikasi domain dan akses VPC untuk Ruang Agen Anda
- Connect GitHub repositori untuk peninjauan kode dan konteks pengujian penetrasi
- (Jika menggunakan Pusat Identitas IAM) Tetapkan pengguna ke Ruang Agen di konsol AWS Security Agent
- (Jika menggunakan IAM-only akses) Luncurkan aplikasi web melalui tautan akses admin di AWS Console
- Buat Ruang Agen tambahan untuk aplikasi lain (lihatthe section called “Buat Ruang Agen”)

Buat Ruang Agen

Buat Ruang Agen untuk mengamankan aplikasi atau proyek di organisasi Anda. Setiap Ruang Agen menyediakan ruang kerja untuk penilaian keamanan, temuan, dan konfigurasi khusus untuk aplikasi atau proyek tersebut dan dapat diakses oleh beberapa pengguna.

Prosedur ini mengasumsikan Anda telah menyelesaikan penyiapan AWS Security Agent awal. Jika Anda belum menyiapkan AWS Security Agent, lihatthe section called “Siapkan Agen Keamanan AWS”.

Gambaran umum

Memahami Ruang Agen

Ruang Agen adalah ruang kerja khusus untuk mengamankan aplikasi atau proyek tertentu. Ini berisi semua ulasan keamanan, repositori kode yang terhubung secara opsional, konfigurasi uji penetrasi, hasil, dan temuan untuk aplikasi itu.

Agen Spaces membantu Anda mengatur pekerjaan keamanan dengan menjaga penilaian keamanan setiap aplikasi terpisah, memungkinkan tim untuk fokus pada aplikasi spesifik mereka. Sebaiknya buat satu Ruang Agen per aplikasi atau proyek untuk mempertahankan batasan yang jelas.

Untuk penjelasan komprehensif tentang Agen Spaces dan bagaimana mereka cocok dengan struktur keamanan organisasi Anda, lihatthe section called “Memahami hierarki sumber daya dan siklus hidup”.

Apa yang termasuk dalam Ruang Agen

Setiap Ruang Agen berisi:

- Repositori kode yang terhubung secara opsional yang terkait dengan aplikasi atau proyek
- Ulasan desain sebelumnya untuk aplikasi atau proyek
- Konfigurasi pengujian penetrasi dan batasan khusus untuk aplikasi
- Hasil tes pengujian penetrasi dan temuan keamanan

Buat Ruang Agen

Buat Ruang Agen baru untuk aplikasi atau proyek yang ingin Anda amankan.

1. Di konsol AWS Security Agent, navigasikan ke halaman Agent Spaces.
2. Pilih Buat Ruang Agen.
3. Di bidang nama Ruang Agen, masukkan nama untuk Ruang Agen Anda.

> **Note**
>
> Nama Ruang Agen ditampilkan kepada pengguna di aplikasi web dan membantu mengidentifikasi aplikasi atau proyek mana yang diwakili oleh ruang ini.

4. (Opsional) Di bidang Deskripsi, berikan deskripsi yang membantu membedakan tujuan Ruang Agen.

> **Tip**
>
> Deskripsi membantu membedakan tujuan Agen Ruang. Sebaiknya jelaskan aplikasi atau proyek spesifik yang akan diamankan oleh Ruang Agen ini, seperti “Aplikasi web portal pelanggan” atau “Layanan mikro pemrosesan pembayaran” atau “Platform analitik internal.”

5. Pilih Buat.

> **Note**
>
> AWS Security Agent akan membuat Ruang Agen Anda. Anda sekarang dapat mengonfigurasi kemampuan dan menghubungkan sumber daya khusus untuk aplikasi ini.

Langkah selanjutnya

Setelah membuat Ruang Agen Anda:

- Connect repositori kode sumber (GitHub,, Bitbucket GitLab, atau GitHub Enterprise Server) untuk peninjauan kode dan konteks pengujian penetrasi
- Connect Confluence untuk konteks dokumentasi dalam tinjauan desain dan pengujian penetrasi
- Aktifkan kemampuan peninjauan kode untuk repositori yang terhubung (lihat) the section called “Aktifkan tinjauan kode permintaan tarik untuk GitHub repositori”
- Konfigurasikan kemampuan pengujian penetrasi termasuk verifikasi domain

- (Jika menggunakan Pusat Identitas IAM) Tetapkan pengguna ke Ruang Agen ini di bawah bagian Aplikasi Web di halaman Ruang Agen. (lihatthe section called “Berikan pengguna akses ke aplikasi web AWS Security Agent”)
- (Jika menggunakan IAM-only akses) Pengguna dengan akses konsol dapat meluncurkan aplikasi web melalui tautan akses admin untuk Ruang Agen ini

Connect integrasi

Connect dan kelola penyedia kode sumber (GitHub, GitLab, Bitbucket, dan GitHub Enterprise Server) dan penyedia dokumentasi (Confluence), bersama dengan konektivitas jaringan pribadi, yang digunakan Agen Spaces Anda untuk peninjauan kode, pengujian penetrasi, dan pemodelan ancaman.

Bagaimana integrasi bekerja dengan Agen Spaces

AWS Security Agent terhubung ke penyedia kode sumber dan dokumentasi pihak ketiga sehingga agen dapat menganalisis kode dan dokumentasi Anda selama penilaian keamanan. Sebelum Anda menghubungkan penyedia tertentu, ada baiknya Anda memahami bagaimana integrasi terkait dengan Ruang Agen dan kemampuan.

Daftar sekali, gunakan kembali di mana saja

Menghubungkan penyedia ke AWS Security Agent melibatkan dua langkah berbeda yang terjadi pada tingkat yang berbeda:

1. Daftarkan integrasi (tingkat akun). Anda mendaftarkan penyedia satu kali dari halaman Integrasi di AWS Security Agent Management Console. Registrasi mewakili koneksi resmi ke akun penyedia, seperti GitHub organisasi, GitLab grup, ruang kerja Bitbucket, atau situs Confluence. Pendaftaran adalah sumber daya tingkat akun.
2. Hubungkan sumber daya ke Agent Space (level Agent Space). Dari dalam Ruang Agen, Anda menghubungkan sumber daya dari integrasi terdaftar — repositori untuk penyedia kode sumber, atau halaman untuk Confluence — dan mengonfigurasi cara agen menggunakannya. Langkah ini khusus untuk setiap Ruang Agen.

Pendaftaran tunggal digunakan kembali di setiap Ruang Agen di akun Anda. Jika Anda mendaftarkan penyedia saat menyiapkan satu Ruang Agen, pendaftaran yang sama tersedia saat Anda

menghubungkan sumber daya ke Ruang Agen lain — Anda tidak mendaftarkan penyedia yang sama lagi.

Satu koneksi, dibagi di seluruh kemampuan

Saat Anda menghubungkan sumber daya ke Ruang Agen, sumber daya tersebut dibagikan di seluruh kemampuan agen. Anda tidak menghubungkan repositori secara terpisah untuk setiap kemampuan.

- Akses baca diberikan dengan menghubungkan sumber daya. Menghubungkan repositori memungkinkan AWS Security Agent membacanya untuk pemindaian kode lengkap (tinjauan kode), konteks pengujian penetrasi, pemodelan ancaman, dan tinjauan desain. Menghubungkan halaman Confluence memungkinkan agen membacanya untuk konteks dokumentasi dalam tinjauan desain, pemodelan ancaman, dan tes penetrasi.
- Tindakan tulis adalah opt-in, per repositori. Saat Anda menghubungkan repositori kode sumber ke Ruang Agen, Anda juga dapat mengaktifkan tindakan tulis untuk setiap repositori:
  - Komentar tinjauan kode - AWS Security Agent memposting temuan ulasan sebagai komentar pada permintaan tarik (atau menggabungkan permintaan GitLab).
  - Remediasi kode - AWS Security Agent membuka permintaan tarik (atau menggabungkan permintaan) dengan perbaikan untuk kerentanan yang ditemukan.

    Tindakan tulis ini tidak tersedia untuk repositori publik. Read-only Analisa masih berlaku.

> **Note**
>
> Confluence adalah penyedia dokumentasi daripada penyedia kode sumber. AWS Security Agent membaca halaman Confluence yang terhubung untuk memberikan konteks tinjauan desain, pemodelan ancaman, dan pengujian penetrasi. Memilih halaman memberikan akses baca; tidak ada sakelar kemampuan per halaman.

Penyedia yang didukung

AWS Security Agent mendukung penyedia berikut:

- Kode sumber - GitHub (GitHub Enterprise yang dihosting cloud GitHub dan yang dihosting di cloud), GitHub Server Perusahaan (di-host sendiri), (dihosting di cloud), GitLab (dihosting sendiri), GitLab Self-Managed dan Bitbucket Cloud.

- Dokumentasi - Confluence Cloud.

Untuk penyedia yang di-host sendiri yang tidak dapat dijangkau melalui internet publik, Anda dapat mengarahkan lalu lintas agen melalui koneksi pribadi. Untuk informasi selengkapnya, lihat the section called “Connect ke kontrol sumber yang dihosting secara pribadi”.

Langkah selanjutnya

- Daftarkan penyedia dari halaman Integrasi. Lihat topik sambungkan untuk penyedia Anda, sepertithe section called “Connect AWS Security Agent ke GitHub repositori”.
- Hubungkan sumber daya ke Ruang Agen dan konfigurasikan kemampuan. Lihat the section called “Aktifkan peninjauan kode” dan the section called “Aktifkan uji penetrasi”.

Connect AWS Security Agent ke GitHub repositori

Hubungkan AWS Security Agent Anda ke GitHub repositori untuk mengaktifkan peninjauan kode, pemodelan ancaman, pengujian penetrasi, dan kemampuan remediasi otomatis. AWS Security Agent mendukung Enterprise yang dihosting di cloud GitHub dan yang dihosting di cloud GitHub . Sebelum memulai, tinjau the section called “Bagaimana integrasi bekerja dengan Agen Spaces” untuk memahami bagaimana pendaftaran digunakan kembali di seluruh Ruang Agen dan dibagikan di seluruh kemampuan.

GitHub integrasi melayani berbagai tujuan:

> **Note**
>
> Halaman ini mencakup cloud-hosted GitHub (github.com) dan Enterprise yang dihosting cloud. GitHub Untuk Server GitHub Perusahaan yang di-hosting sendiri, lihatthe section called “Connect AWS Security Agent ke GitHub Enterprise Server”.

- Peninjauan kode - Secara otomatis menganalisis perubahan kode di setiap permintaan tarik terhadap persyaratan keamanan organisasi Anda, dan jalankan pemindaian repositori penuh sesuai permintaan
- Pemodelan ancaman - Memberikan pemahaman aplikasi dengan menganalisis kode sumber, aliran data, dan arsitektur

- Konteks pengujian penetrasi - Memberikan pemahaman aplikasi untuk pengujian penetrasi dengan menganalisis kode sumber
- Remediasi otomatis - Kirim permintaan tarik dengan perbaikan untuk kerentanan yang ditemukan selama penilaian keamanan

Menyambung GitHub ke AWS Security Agent memerlukan otorisasi GitHub Aplikasi AWS Security Agent untuk GitHub organisasi atau akun pengguna Anda, lalu mendaftarkan koneksi di AWS Console.

Bagaimana GitHub integrasi bekerja

Analisis permintaan tarik terjadi di dalam GitHub. Setelah Anda mengotorisasi GitHub Aplikasi, menghubungkan repositori, dan mengaktifkan komentar peninjauan kode di AWS Management Console, AWS Security Agent secara otomatis memindai perubahan di setiap permintaan tarik baru (pemindaian diferensial hanya kode yang diubah) dan memposting temuan sebagai komentar permintaan tarik.

Anda membuat dan menjalankan tinjauan kode lengkap — yang memindai seluruh basis kode repositori — di aplikasi web AWS Security Agent, bukan di aplikasi web AWS Security Agent. GitHub

Pengujian penetrasi dan pemodelan ancaman dimulai dalam Aplikasi Web AWS Security Agent. Pengguna memilih repositori yang terhubung untuk menyediakan konteks aplikasi, dan menentukan domain target untuk pengujian penetrasi. Jika Anda mengaktifkan remediasi otomatis, pengguna dapat meminta AWS Security Agent untuk memperbaiki temuan dengan membuka permintaan tarik ke repositori yang terhubung.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- GitHub akses admin organisasi atau akses pemilik akun GitHub pengguna
- Izin untuk mengonfigurasi integrasi untuk Ruang Agen Anda di AWS Management Console
- Memahami repositori mana yang ingin Anda sambungkan untuk peninjauan kode, pemodelan ancaman, dan pengujian penetrasi

Important

GitHub Aplikasi hanya dapat diinstal sekali ke GitHub akun atau GitHub organisasi. Jika Anda perlu menghubungkan GitHub organisasi yang sama ke AWS Security Agent, Anda harus menggunakan akun AWS yang sama tempat integrasi pertama kali didaftarkan.

Note

Jika organisasi GitHub perusahaan Anda telah mengaktifkan daftar yang diizinkan IP, Anda harus menerima alamat IP yang diizinkan di GitHub aplikasi. Anda juga dapat memilih untuk secara otomatis menambahkan alamat IP ke daftar izin Anda. Untuk informasi selengkapnya, lihat Mengizinkan akses oleh GitHub Aplikasi dan Mengaktifkan alamat IP yang diizinkan dalam GitHub dokumentasi.
Alamat IP berikut digunakan untuk mengakses GitHub sumber daya Anda:

- US East (N. Virginia) (us-east-1)
  - `34.228.181.128`
  - `44.219.176.187`
  - `54.226.244.221`
- US West (Oregon) (us-west-2)
  - `34.212.16.133`
  - `52.89.67.212`
  - `54.187.135.61`
- Asia Pasifik (Mumbai) (ap-south-1)
  - `13.126.209.199`
  - `13.234.6.24`
  - `35.154.102.216`
- Asia Pasifik (Singapura) (ap-southeast-1)
  - `18.139.13.125`
  - `47.130.240.215`
  - `54.179.238.173`
- Asia Pacific (Sydney) (ap-southeast-2)

    - 13.237.95.197
    - 13.238.84.102
    - 52.64.174.242
  - Asia Pacific (Tokyo) (ap-northeast-1)
    - 13.192.12.233
    - 35.74.181.230
    - 57.183.50.158
  - Eropa (Frankfurt) (eu-central-1)
    - 18.158.110.140
    - 52.57.96.160
    - 52.59.55.56
  - Eropa (Irlandia) (eu-west-1)
    - 34.251.85.24
    - 52.30.157.157
    - 52.51.192.222
  - Amerika Selatan (Sao Paulo) (sa-east-1)
    - 54.94.247.213
    - 54.207.222.14
    - 54.232.201.242

Otorisasi dan daftarkan Aplikasi AWS Security Agent GitHub

Otorisasi GitHub Aplikasi AWS Security Agent untuk mengakses GitHub organisasi atau akun pengguna Anda, lalu daftarkan koneksi di AWS Console.

> **Important**
>
> Selesaikan semua langkah dalam proses ini tanpa menutup browser Anda atau menavigasi. Jika proses pendaftaran terganggu, Anda mungkin perlu mencopot pemasangan GitHub Aplikasi dan memulai dari awal.

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.

2. Pilih Tambahkan integrasi.
3. Pilih GitHub.
4. Pilih Berikutnya.
5. Pilih Instal dan otorisasi.

   Anda dialihkan GitHub untuk menyelesaikan otorisasi. Pastikan Anda masuk GitHub dengan akun yang memiliki akses admin ke organisasi atau akun pengguna yang ingin Anda sambungkan.
6. Di GitHub, pilih akun atau organisasi tempat Anda ingin menginstal GitHub Aplikasi AWS Security Agent.
7. Pilih repositori AWS Security Agent mana yang dapat diakses:
   - Semua repositori - Berikan akses ke semua repositori saat ini dan masa depan di organisasi atau akun pengguna
   - Hanya pilih repositori - Pilih repositori tertentu dari dropdown. Anda dapat memilih beberapa repositori satu per satu.

   > **Note**
   >
   > Anda dapat mengubah akses repositori kapan saja dengan mengunjungi pengaturan GitHub Aplikasi di pengaturan GitHub organisasi atau akun pengguna Anda.
8. Pilih Instal dan otorisasi.
9. Anda diarahkan kembali ke AWS Management Console untuk menyelesaikan pendaftaran.
10. Di bagian Detail pendaftaran, konfigurasikan bidang berikut:
    a. Nama pendaftaran - Masukkan nama deskriptif untuk GitHub koneksi ini. Gunakan nama yang mengidentifikasi GitHub organisasi atau akun pengguna, seperti "" atau Acme-Corp-Org "Production-Repos”.
    b. Jenis akun - Pilih salah satu dari berikut ini dari dropdown:
       - Organisasi - Jika Anda menghubungkan akun GitHub organisasi
       - Pengguna - Jika Anda menghubungkan akun GitHub pengguna pribadi
    c. Nama organisasi (hanya muncul jika Anda memilih Organisasi) - Masukkan nama persis GitHub organisasi Anda seperti yang muncul di GitHub.
11. Pilih Hubungkan.

12 Anda melihat pesan konfirmasi dan kembali ke halaman Integrasi, di mana GitHub koneksi baru Anda muncul dengan nama pendaftarannya. Untuk menghubungkan GitHub organisasi atau akun pengguna tambahan, ulangi proses ini dengan memilih Tambah integrasi lagi.

Memecahkan masalah integrasi GitHub

Jika Anda mengalami masalah selama proses GitHub integrasi, gunakan panduan berikut untuk menyelesaikan masalah umum.

Tidak dapat menyelesaikan pendaftaran

Jika Anda tidak dapat menyelesaikan proses pendaftaran (misalnya, browser Anda ditutup, Anda menavigasi dari halaman pendaftaran, atau mengalami gangguan sesi), GitHub Aplikasi AWS Security Agent dapat diinstal di GitHub organisasi Anda tetapi tidak terdaftar di AWS Console.

Gejala:

- Saat Anda mencoba mengotorisasi GitHub Aplikasi lagi, tampilkan GitHub “Konfigurasi” alih-alih “Instal”
- Anda tidak dapat menyelesaikan pendaftaran di AWS Console
- Integrasi tidak muncul dalam daftar Integrasi Anda

Resolusi:

1. Copot pemasangan GitHub Aplikasi AWS Security Agent dari GitHub organisasi atau akun pengguna Anda.
2. Kembali ke konsol AWS Security Agent dan mulai proses integrasi lagi dari awal.

Beberapa akun AWS mencoba mengintegrasikan GitHub organisasi yang sama

GitHub Aplikasi hanya dapat diinstal sekali ke GitHub akun atau GitHub organisasi. Jika Anda perlu menggunakan repositori dari GitHub organisasi yang sudah terintegrasi dengan akun AWS Security Agent yang berbeda, Anda harus menggunakan akun AWS tempat integrasi pertama kali didaftarkan.

Resolusi:

- Identifikasi akun AWS Security Agent mana yang memiliki GitHub integrasi terdaftar
- Gunakan akun AWS tersebut untuk membuat Ruang Agen dan menghubungkan repositori

- Jika Anda perlu memindahkan integrasi ke akun AWS yang berbeda, hapus instalan GitHub Aplikasi dari akun AWS asli terlebih dahulu, lalu integrasikan dengan akun baru

Langkah selanjutnya

Setelah terhubung GitHub ke AWS Security Agent:

- Arahkan ke Ruang Agen tempat Anda ingin menggunakan repositori ini
- Pilih Aktifkan tinjauan kode atau Setup pengujian penetrasi untuk menghubungkan repositori tertentu ke Ruang Agen Anda dan mengonfigurasi penggunaannya
- Aktifkan remediasi otomatis untuk memungkinkan AWS Security Agent mengirimkan permintaan tarik dengan perbaikan kerentanan
- Tinjau Izin GitHub aplikasi dan akses repositori di setelan organisasi Anda GitHub

Hapus GitHub integrasi

Hapus GitHub integrasi saat Anda tidak lagi memerlukan AWS Security Agent untuk mengakses repositori dari GitHub organisasi atau akun pengguna tertentu. Anda harus terlebih dahulu menghapus GitHub aplikasi AWS Security Agent GitHub sebelum menghapus integrasi di AWS Console.

Prasyarat untuk penghapusan

Sebelum menghapus GitHub integrasi, pastikan Anda memiliki:

- Memeriksa Ruang Agen mana yang memiliki repositori yang terhubung dari integrasi ini
- Memahami dampaknya: Menghapus integrasi ini akan memutus koneksi repositori untuk peninjauan kode, konteks pengujian penetrasi, dan remediasi uji penetrasi di semua Ruang Agen menggunakan repositori dari integrasi ini
- GitHub akses admin organisasi atau akses pemilik akun pengguna untuk menghapus instalasi Aplikasi GitHub

Langkah 1: Copot pemasangan GitHub Aplikasi AWS Security Agent dari GitHub

Pertama, hapus instalan GitHub Aplikasi AWS Security Agent dari GitHub organisasi atau akun pengguna Anda.

Untuk GitHub Organizations:

1. Buka github.com dan buka halaman organisasi Anda.
2. Di bilah sisi kiri, pilih Pengaturan.
3. Di bawah Kode, perencanaan, dan otomatisasi, pilih GitHub Aplikasi Terinstal.
4. Temukan aplikasi AWS Security Agent dalam daftar.
5. Pilih Konfigurasi di samping aplikasi AWS Security Agent.
6. Gulir ke bagian bawah halaman konfigurasi dan pilih Uninstall.
7. Konfirmasikan penghapusan instalasi saat diminta.

Untuk Akun GitHub Pengguna:

1. Pergi ke github.com.
2. Pilih gambar profil Anda.
3. Pilih Pengaturan.
4. Di bilah sisi kiri, pilih Aplikasi.
5. Buka tab GitHub Aplikasi Terinstal.
6. Temukan aplikasi AWS Security Agent dalam daftar.
7. Pilih Konfigurasi di samping aplikasi AWS Security Agent.
8. Gulir ke bagian bawah halaman konfigurasi dan pilih Uninstall.
9. Konfirmasikan penghapusan instalasi saat diminta.

Langkah 2: Hapus integrasi dari AWS Security Agent

Setelah mencopot pemasangan GitHub Aplikasi, hapus pendaftaran integrasi dari AWS Console.

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Temukan GitHub integrasi yang ingin Anda hapus dalam daftar integrasi.
3. Pilih integrasi dengan mengkliknya.
4. Pilih Hapus.
5. Tinjau dialog konfirmasi, yang memperingatkan Anda tentang dampaknya:

> **Warning**
>
> Menghapus integrasi ini akan memengaruhi semua Ruang Agen yang memiliki repositori yang terhubung dari GitHub organisasi atau akun pengguna ini. Ini akan rusak:
>
> - Fungsionalitas peninjauan kode untuk repositori yang terhubung
> - Konteks pengujian penetrasi dari repositori yang terhubung
> - Kemampuan remediasi uji penetrasi untuk repositori yang terhubung
>
> Pastikan Anda telah menghapus GitHub aplikasi AWS Security Agent GitHub sebelum melanjutkan.

6. Jika Anda belum menghapus GitHub aplikasi dari GitHub, Anda akan menerima peringatan. Kembali ke Langkah 1 untuk menyelesaikan penghapusan instalasi terlebih dahulu.
7. Jika Anda telah menghapus GitHub aplikasi dan memahami dampaknya, pilih Konfirmasi penghapusan.
8. Integrasi dihapus dari daftar integrasi Anda. Setiap Ruang Agen dengan repositori dari integrasi ini tidak lagi memiliki akses ke repositori tersebut untuk peninjauan kode, konteks pengujian penetrasi, atau remediasi otomatis.

Connect AWS Security Agent ke GitLab repositori

Hubungkan AWS Security Agent Anda ke repositori GitLab Cloud untuk mengaktifkan peninjauan kode, pemodelan ancaman, pengujian penetrasi, dan kemampuan remediasi otomatis. Sebelum memulai, tinjau the section called “Bagaimana integrasi bekerja dengan Agen Spaces” untuk memahami bagaimana pendaftaran digunakan kembali di seluruh Ruang Agen dan dibagikan di seluruh kemampuan.

GitLab integrasi melayani berbagai tujuan:

- Peninjauan kode - Secara otomatis menganalisis perubahan kode di setiap permintaan gabungan terhadap persyaratan keamanan organisasi Anda, dan jalankan pemindaian repositori penuh sesuai permintaan
- Pemodelan ancaman - Memberikan pemahaman aplikasi dengan menganalisis kode sumber, aliran data, dan arsitektur
- Konteks pengujian penetrasi - Memberikan pemahaman aplikasi untuk pengujian penetrasi dengan menganalisis kode sumber

- Remediasi otomatis - Kirim permintaan gabungan dengan perbaikan untuk kerentanan yang ditemukan selama penilaian keamanan

Menyambung GitLab ke AWS Security Agent memerlukan penyediaan token GitLab akses dengan izin yang sesuai, lalu mendaftarkan koneksi di AWS Console.

Bagaimana GitLab integrasi bekerja

Analisis permintaan gabungan terjadi di dalam GitLab. Setelah Anda memberikan token akses, menghubungkan proyek, dan mengaktifkan komentar tinjauan kode di AWS Management Console, AWS Security Agent secara otomatis memindai perubahan di setiap permintaan penggabungan baru (pemindaian diferensial hanya kode yang diubah) dan memposting temuan sebagai komentar permintaan gabungan.

Anda membuat dan menjalankan tinjauan kode lengkap — yang memindai seluruh basis kode proyek — di aplikasi web AWS Security Agent, bukan di GitLab aplikasi web AWS Security Agent.

Pengujian penetrasi dan pemodelan ancaman dimulai dalam Aplikasi Web AWS Security Agent. Pengguna menentukan domain target dan memilih repositori yang terhubung untuk menyediakan konteks aplikasi. Jika Anda mengaktifkan remediasi otomatis, pengguna dapat meminta AWS Security Agent untuk memperbaiki temuan dengan membuka permintaan gabungan ke repositori yang terhubung.

> **Note**
>
> Remediasi otomatis tidak tersedia untuk GitLab repositori publik untuk menghindari pengungkapan kerentanan sebelum diperbaiki.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- GitLab.com Akun dengan akses Pengelola atau Pemilik ke proyek yang ingin Anda sambungkan
- Token GitLab akses dengan cakupan yang diperlukan untuk jenis koneksi Anda:
  - Pribadi - Token akses pribadi dengan semua izin baca dan `api` izin.
  - Grup - Token akses grup dengan `read_api` dan `read_repository` cakupan.
- Izin untuk mengonfigurasi integrasi di AWS Security Agent Management Console

> **Important**
>
> Setel kedaluwarsa token ke maksimum 365 hari setelah tanggal saat ini.

> **Note**
>
> GitLab Token Akses Pribadi dapat digunakan di beberapa akun AWS. Tidak seperti GitHub integrasi Atlassian, tidak ada batasan untuk menghubungkan akun yang sama GitLab ke beberapa instans AWS Security Agent.

Daftarkan GitLab koneksi

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Pilih Tambahkan integrasi.
3. Pilih GitLab, lalu pilih Berikutnya.
4. Di bawah Pilih jenis akun, pilih salah satu dari berikut ini:
   - Personal - Hubungkan akun GitLab pengguna individual Anda.
   - Grup - Connect GitLab grup yang berisi beberapa proyek. Masukkan ID Grup Anda.
5. Di bidang Access token, tempel token GitLab akses Anda.
6. Di bidang Nama pendaftaran, masukkan nama deskriptif untuk koneksi ini, seperti`Engineering-Team-GitLab`. Karakter yang valid adalah huruf, angka, titik, garis bawah, dan tanda hubung.
7. Pilih Hubungkan.

   Anda kembali ke halaman Integrasi, di mana koneksi baru muncul dengan nama pendaftarannya.

Memecahkan masalah integrasi GitLab

Jika Anda mengalami masalah selama proses GitLab integrasi, gunakan panduan berikut untuk menyelesaikan masalah umum.

Token tidak valid atau kedaluwarsa

Gejala

- Integrasi gagal terhubung

- Integrasi kerja sebelumnya berhenti berfungsi
- Pesan kesalahan yang menunjukkan kegagalan otentikasi

Resolusi

1. Masuk GitLab, navigasikan ke pengaturan pengguna Anda dan pilih Token Akses.
2. Verifikasi token Anda belum kedaluwarsa.
3. Verifikasi token memiliki cakupan yang diperlukan untuk jenisnya.
4. Jika token telah kedaluwarsa, buat token baru dan perbarui integrasi di AWS Console.

Pembatasan tarif

Gejala

- Kegagalan intermiten selama peninjauan kode
- Analisis permintaan penggabungan tertunda

Resolusi

- GitLab menerapkan batas tarif untuk permintaan API. Jika Anda memiliki sejumlah besar repositori atau permintaan gabungan yang sering, beberapa permintaan mungkin dibatasi.
- Tunggu hingga jendela batas tarif diatur ulang, atau hubungi GitLab dukungan untuk meningkatkan batas tarif Anda.

Langkah selanjutnya

Setelah terhubung GitLab ke AWS Security Agent:

- Arahkan ke Ruang Agen tempat Anda ingin menggunakan repositori ini
- Pilih Aktifkan tinjauan kode atau Setup pengujian penetrasi untuk menghubungkan proyek tertentu ke Ruang Agen Anda dan mengonfigurasi penggunaannya
- Aktifkan remediasi Kode untuk memungkinkan AWS Security Agent mengirimkan permintaan gabungan dengan perbaikan kerentanan
- Untuk GitLab instans yang dihosting secara pribadi, lihat the section called “Connect AWS Security Agent ke GitLab Self-Managed”

Connect AWS Security Agent ke GitLab Self-Managed

Hubungkan AWS Security Agent Anda ke GitLab Self-Managed instans untuk mengaktifkan peninjauan kode, pemodelan ancaman, pengujian penetrasi, dan kemampuan remediasi otomatis untuk repositori yang dihosting di infrastruktur Anda sendiri.

GitLab Self-Managed integrasi bekerja sama seperti GitLab Cloud (lihatthe section called “Connect AWS Security Agent ke GitLab repositori”) dengan konfigurasi tambahan untuk konektivitas jaringan ke instans pribadi Anda. Sebelum memulai, tinjau the section called “Bagaimana integrasi bekerja dengan Agen Spaces” untuk memahami bagaimana pendaftaran digunakan kembali di seluruh Ruang Agen dan dibagikan di seluruh kemampuan.

> **Note**
>
> GitLab Self-Managed terdaftar melalui GitLabintegrasi, bukan jenis integrasi terpisah. Dalam alur pendaftaran, Anda memilih Gunakan titik akhir yang GitLab dihosting sendiri dan berikan URL instans Anda. GitLab Alur yang dihosting cloud dijelaskan dalam. the section called “Connect AWS Security Agent ke GitLab repositori”

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Sebuah GitLab Self-Managed contoh yang baik:
  - Dapat diakses publik melalui internet, ATAU
  - Dapat diakses melalui koneksi pribadi (lihatthe section called “Connect ke kontrol sumber yang dihosting secara pribadi”)
- Token GitLab akses dengan cakupan yang diperlukan untuk jenis koneksi Anda:
  - Pribadi - Token akses pribadi dengan semua izin baca dan `api` izin.
  - Grup - Token akses grup dengan `read_api` dan `read_repository` cakupan.
- Akses pengelola atau Pemilik ke proyek yang ingin Anda sambungkan
- GitLab Instans Anda harus melayani lalu lintas HTTPS dengan versi TLS minimum 1.2

> **Note**
>
> Jika GitLab Self-Managed instans Anda menggunakan sertifikat TLS yang dikeluarkan oleh otoritas sertifikat pribadi, Anda dapat memberikan kunci PEM-encoded publik sertifikat saat membuat koneksi pribadi. Hal ini memungkinkan AWS Security Agent mempercayai koneksi TLS ke instans Anda.

Daftarkan GitLab Self-Managed koneksi

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Pilih Tambahkan integrasi.
3. Pilih GitLab, lalu pilih Berikutnya.
4. Di bawah Pilih jenis akun, pilih Pribadi atau Grup. Jika Anda memilih Grup, masukkan ID Grup Anda.
5. Pilih Gunakan titik akhir yang GitLab dihosting sendiri.
6. Di bidang URL titik akhir yang GitLab dihosting sendiri, masukkan URL instance Anda, misalnya. `https://gitlab.example.com`
7. Jika instans Anda tidak dapat diakses publik, pilih Connect to endpoint menggunakan koneksi pribadi, lalu pilih koneksi pribadi yang ada atau buat yang baru. Lihat the section called “Connect ke kontrol sumber yang dihosting secara pribadi”.
8. Di bidang Access token, tempel token GitLab akses Anda.
9. Di bidang Nama pendaftaran, masukkan nama deskriptif untuk koneksi ini. Karakter yang valid adalah huruf, angka, titik, garis bawah, dan tanda hubung.
10. Pilih Hubungkan.

    Anda kembali ke halaman Integrasi, di mana koneksi baru muncul dengan nama pendaftarannya.

Konektivitas pribadi

Jika GitLab Self-Managed instans Anda tidak dapat diakses publik, Anda harus membuat koneksi pribadi sebelum mendaftarkan integrasi. Lihat the section called “Connect ke kontrol sumber yang dihosting secara pribadi” untuk instruksi terperinci.

> **Important**
>
> Service-managed koneksi pribadi mengharuskan GitLab Self-Managed instance berjalan di akun AWS yang sama di mana Ruang Agen dibuat. Untuk akses lintas akun, gunakan koneksi pribadi yang dikelola sendiri di mana Anda menyediakan konfigurasi sumber daya VPC Lattice Anda sendiri.

Memecahkan masalah integrasi GitLab Self-Managed

Selain langkah-langkah pemecahan masalahthe section called “Connect AWS Security Agent ke GitLab repositori”, masalah berikut khusus untuk instans yang dikelola sendiri:

Instance tidak dapat dijangkau

Gejala

- Koneksi gagal dengan batas waktu atau kesalahan jaringan
- Integrasi sebelumnya berfungsi tetapi berhenti berfungsi

Resolusi

- Verifikasi GitLab instans Anda berjalan dan dapat diakses
- Jika menggunakan koneksi pribadi, verifikasi gateway sumber daya VPC Lattice sehat dan ENI memiliki konektivitas jaringan ke instans Anda
- Verifikasi grup keamanan mengizinkan lalu lintas pada port yang dikonfigurasi
- Verifikasi sertifikat TLS valid dan tidak kedaluwarsa

Kesalahan sertifikat TLS

Gejala

- Koneksi gagal dengan SSL/TLS kesalahan

Resolusi

- Verifikasi instans Anda menyajikan HTTPS dengan TLS 1.2 atau lebih tinggi

- Jika menggunakan otoritas sertifikat pribadi, pastikan kunci PEM-encoded publik disediakan selama pengaturan koneksi pribadi
- Verifikasi sertifikat tidak kedaluwarsa

Langkah selanjutnya

Setelah terhubung GitLab Self-Managed ke AWS Security Agent:

- Arahkan ke Ruang Agen tempat Anda ingin menggunakan repositori ini
- Pilih Aktifkan tinjauan kode atau Setup pengujian penetrasi untuk menghubungkan proyek tertentu
- Aktifkan remediasi Kode untuk menggabungkan perbaikan berbasis permintaan

Hapus GitLab integrasi

Hapus GitLab integrasi saat Anda tidak lagi memerlukan AWS Security Agent untuk mengakses repositori dari GitLab akun atau grup tertentu.

Prasyarat untuk penghapusan

Sebelum menghapus GitLab integrasi, pastikan Anda memiliki:

- Memeriksa Ruang Agen mana yang memiliki repositori yang terhubung dari integrasi ini
- Memahami dampaknya: Menghapus integrasi ini akan memutus koneksi repositori untuk peninjauan kode, konteks pengujian penetrasi, pemodelan ancaman, dan remediasi otomatis di semua Ruang Agen menggunakan repositori dari integrasi ini

Hapus integrasi dari AWS Security Agent

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Temukan GitLab integrasi yang ingin Anda hapus.
3. Pilih integrasi dengan mengkliknya.
4. Pilih Hapus.
5. Tinjau dialog konfirmasi dan pilih Konfirmasi penghapusan.

> **Note**
>
> Setelah menghapus integrasi, Anda mungkin juga ingin mencabut Token Akses Pribadi GitLab untuk mencegah akses lebih lanjut. Arahkan ke pengaturan GitLab pengguna Anda dan hapus atau cabut token.

Proses yang sama berlaku untuk GitLab Self-Managed integrasi.

Connect AWS Security Agent ke GitHub Enterprise Server

Hubungkan AWS Security Agent Anda ke instans GitHub Enterprise Server (GHES) untuk mengaktifkan peninjauan kode, pemodelan ancaman, pengujian penetrasi, dan kemampuan remediasi otomatis untuk repositori yang dihosting di infrastruktur Anda sendiri.

GitHub Integrasi Enterprise Server menyediakan kemampuan yang sama seperti yang dihosting di cloud GitHub (lihat Connect AWS Security Agent ke GitHub repositori) dengan konfigurasi tambahan untuk konektivitas jaringan ke instans yang dihosting sendiri. Sebelum memulai, tinjau the section called “Bagaimana integrasi bekerja dengan Agen Spaces” untuk memahami bagaimana pendaftaran digunakan kembali di seluruh Ruang Agen dan dibagikan di seluruh kemampuan.

> **Note**
>
> GitHub Enterprise Server terdaftar melalui GitHubintegrasi, bukan jenis integrasi terpisah. Dalam alur pendaftaran Anda memilih GitHub Enterprise Server sebagai jenis instans. GitHub.com Alur yang dihosting cloud dijelaskan dalam. the section called “Connect AWS Security Agent ke GitHub repositori”

Cara kerja integrasi GitHub Enterprise Server

Analisis permintaan tarik terjadi di dalam Server GitHub Perusahaan. Setelah Anda mengotorisasi koneksi, menghubungkan repositori, dan mengaktifkan komentar tinjauan kode di AWS Management Console, AWS Security Agent secara otomatis memindai perubahan di setiap permintaan tarik baru (pemindaian diferensial hanya kode yang diubah) dan memposting temuan sebagai komentar permintaan tarik.

Anda membuat dan menjalankan tinjauan kode lengkap — yang memindai seluruh basis kode repositori — di aplikasi web AWS Security Agent, bukan di GitHub Enterprise Server.

Pengujian penetrasi dan pemodelan ancaman dimulai dalam Aplikasi Web AWS Security Agent. Pengguna menentukan domain target dan memilih repositori yang terhubung untuk menyediakan konteks aplikasi. Jika Anda mengaktifkan remediasi otomatis, pengguna dapat meminta AWS Security Agent untuk memperbaiki temuan dengan membuka permintaan tarik ke repositori yang terhubung.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Sebuah contoh GitHub Enterprise Server yang baik:
  - Dapat diakses publik melalui internet, ATAU
  - Dapat diakses melalui koneksi pribadi (lihatthe section called “Connect ke kontrol sumber yang dihosting secara pribadi”)
- Administrator situs atau akses administrator organisasi pada instans GHES Anda
- Instans GHES Anda harus melayani lalu lintas HTTPS dengan versi TLS minimum 1.2
- Izin untuk mengonfigurasi integrasi di AWS Security Agent Management Console

> **Note**
>
> GitHub Integrasi Enterprise Server dapat digunakan di beberapa akun AWS.

Daftarkan koneksi GitHub Enterprise Server

Mendaftarkan koneksi GitHub Enterprise Server menggunakan alur OAuth-based otorisasi.

> **Important**
>
> Selesaikan semua langkah dalam proses ini tanpa menutup browser Anda atau menavigasi. Jika proses pendaftaran terganggu, Anda mungkin perlu memulai ulang dari awal.

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Pilih Tambahkan integrasi.
3. Pilih GitHub, lalu pilih Berikutnya.

4. Di bawah Jenis instans, pilih Server GitHub Perusahaan.
5. Di bidang URL Server GitHub Perusahaan, masukkan URL HTTPS instance Anda, misalnya`https://github.example.com`.
6. Jika instans Anda tidak dapat diakses publik, pilih Connect to endpoint menggunakan koneksi pribadi, lalu pilih koneksi pribadi yang ada atau buat yang baru. Lihat the section called “Connect ke kontrol sumber yang dihosting secara pribadi”.
7. Di bagian Register details, konfigurasikan bidang-bidang berikut:
   a. Nama pendaftaran - Masukkan nama deskriptif untuk koneksi ini. Karakter yang valid adalah huruf, angka, titik, garis bawah, dan tanda hubung.
   b. GitHub jenis akun - Pilih Organisasi atau Pengguna.
   c. Nama organisasi (hanya muncul jika Anda memilih Organisasi) - Masukkan nama yang tepat dari organisasi Server GitHub Perusahaan Anda. Nilai peka huruf besar/kecil.
8. Pilih Hubungkan.

> **Note**
>
> AWS Security Agent mengalihkan Anda dari konsol untuk menyelesaikan otorisasi dengan instans Server GitHub Perusahaan Anda. Setelah otorisasi selesai, Anda kembali ke konsol dan koneksi baru muncul di halaman Integrasi.

Konektivitas pribadi

Jika instance GitHub Enterprise Server Anda tidak dapat diakses publik, Anda harus membuat koneksi pribadi sebelum mendaftarkan integrasi. Lihat the section called “Connect ke kontrol sumber yang dihosting secara pribadi” untuk instruksi terperinci.

> **Important**
>
> Service-managed koneksi pribadi memerlukan instans GHES untuk berjalan di akun AWS yang sama di mana Ruang Agen dibuat. Untuk akses lintas akun, gunakan koneksi pribadi yang dikelola sendiri.

> **Note**
>
> Jika instans GHES Anda menggunakan sertifikat TLS yang dikeluarkan oleh otoritas sertifikat pribadi, berikan kunci PEM-encoded publik sertifikat saat membuat koneksi pribadi.

Memecahkan masalah integrasi Server GitHub Perusahaan

Jika Anda mengalami masalah saat menghubungkan AWS Security Agent ke GitHub Enterprise Server, gunakan panduan berikut untuk mendiagnosis dan menyelesaikan masalah umum.

Kegagalan pengalihan OAuth

Gejala

- Pengalihan browser gagal selama alur otorisasi
- Halaman kesalahan ditampilkan setelah otorisasi pada GHES

Resolusi

- Verifikasi instans GHES Anda dapat diakses dari browser Anda
- Pastikan URL callback OAuth dikonfigurasi dengan benar
- Mulai ulang proses integrasi dari awal

Instance tidak dapat dijangkau

Gejala

- Koneksi gagal dengan batas waktu atau kesalahan jaringan

Resolusi

- Verifikasi instans GHES Anda berjalan dan dapat diakses
- Jika menggunakan koneksi pribadi, verifikasi konektivitas VPC Lattice
- Verifikasi grup keamanan mengizinkan lalu lintas pada port yang dikonfigurasi
- Verifikasi sertifikat TLS valid (minimal TLS 1.2)

Langkah selanjutnya

Setelah menghubungkan GitHub Enterprise Server ke AWS Security Agent:

- Arahkan ke Ruang Agen tempat Anda ingin menggunakan repositori ini
- Pilih Aktifkan tinjauan kode atau Setup pengujian penetrasi untuk menghubungkan repositori tertentu
- Aktifkan remediasi Kode untuk memungkinkan AWS Security Agent mengirimkan permintaan tarik dengan perbaikan kerentanan

Menghapus integrasi GitHub Enterprise Server

Hapus integrasi GitHub Enterprise Server saat Anda tidak lagi memerlukan AWS Security Agent untuk mengakses repositori dari instans GHES tertentu.

Prasyarat untuk penghapusan

Sebelum menghapus integrasi GHES:

- Periksa Ruang Agen mana yang memiliki repositori yang terhubung dari integrasi ini
- Memahami dampaknya: Menghapus akan merusak tinjauan kode, konteks pengujian penetrasi, pemodelan ancaman, dan remediasi otomatis untuk semua repositori yang terhubung

Hapus integrasi

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Temukan integrasi GitHub Enterprise Server yang ingin Anda hapus.
3. Pilih integrasi.
4. Pilih Hapus.
5. Tinjau dialog konfirmasi dan pilih Konfirmasi penghapusan.

> **Note**
>
> Setelah penghapusan, Anda mungkin juga ingin mencabut aplikasi OAuth pada instance GHES Anda. Arahkan ke pengaturan organisasi GHES Anda dan hapus aplikasi resmi.

Temukan ID instalasi Atlassian Anda

Sebelum mendaftarkan integrasi Bitbucket atau Confluence di AWS Console, Anda perlu menemukan ID instalasi dari aplikasi AWS Security Agent Forge yang diinstal di situs Atlassian Anda. Anda menempelkan ID ini ke dalam alur pendaftaran.

Untuk Bitbucket

1. Di Bitbucket, navigasikan ke pengaturan Workspace Anda.
2. Pilih Forge Apps dari sidebar.
3. Temukan Connect with AWS Security Agent dalam daftar aplikasi yang diinstal.
4. Pilih Salin ke Papan Klip untuk menyalin ID instalasi.

Untuk Confluence

1. Di Confluence, arahkan ke administrasi Confluence.
2. Pilih Aplikasi dari sidebar.
3. Temukan Connect with AWS Security Agent dalam daftar aplikasi yang diinstal.
4. Pilih Salin ke Papan Klip untuk menyalin ID instalasi.

Anda akan memerlukan ID penginstalan ini saat menyelesaikan pendaftaran di AWS Security Agent Management Console.

Connect AWS Security Agent ke repositori Bitbucket

Hubungkan AWS Security Agent Anda ke repositori Bitbucket Cloud untuk mengaktifkan peninjauan kode, pemodelan ancaman, pengujian penetrasi, dan kemampuan remediasi otomatis. Sebelum memulai, tinjau the section called “Bagaimana integrasi bekerja dengan Agen Spaces” untuk memahami bagaimana pendaftaran digunakan kembali di seluruh Ruang Agen dan dibagikan di seluruh kemampuan.

Integrasi Bitbucket melayani berbagai tujuan:

- Peninjauan kode - Secara otomatis menganalisis perubahan kode di setiap permintaan tarik terhadap persyaratan keamanan organisasi Anda, dan jalankan pemindaian repositori penuh sesuai permintaan

- Pemodelan ancaman - Memberikan pemahaman aplikasi dengan menganalisis kode sumber, aliran data, dan arsitektur
- Konteks pengujian penetrasi - Memberikan pemahaman aplikasi untuk pengujian penetrasi
- Remediasi otomatis - Kirim permintaan tarik dengan perbaikan untuk kerentanan yang ditemukan selama penilaian keamanan

Menghubungkan Bitbucket ke AWS Security Agent memerlukan penginstalan aplikasi AWS Security Agent Forge di situs Atlassian Anda dan menyelesaikan alur otorisasi OAuth.

> **Note**
>
> AWS Security Agent hanya mendukung Bitbucket Cloud. Bitbucket Server dan Bitbucket Data Center tidak didukung.

Cara kerja integrasi Bitbucket

Analisis permintaan tarik terjadi dalam Bitbucket. Setelah Anda menginstal aplikasi Forge, menghubungkan repositori, dan mengaktifkan komentar tinjauan kode di AWS Management Console, AWS Security Agent secara otomatis memindai perubahan di setiap permintaan tarik baru (pemindaian diferensial hanya kode yang diubah) dan memposting temuan sebagai komentar permintaan tarik.

Anda membuat dan menjalankan ulasan kode lengkap — yang memindai seluruh basis kode repositori — di aplikasi web AWS Security Agent, bukan di Bitbucket.

Pengujian penetrasi dan pemodelan ancaman dimulai dalam Aplikasi Web AWS Security Agent. Pengguna menentukan domain target dan memilih repositori yang terhubung untuk menyediakan konteks aplikasi.

> **Note**
>
> Remediasi otomatis tidak tersedia untuk repositori Bitbucket publik untuk menghindari pengungkapan kerentanan sebelum diperbaiki.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Ruang kerja Bitbucket Cloud dengan akses admin
- Akun Atlassian dengan hak administrator situs
- Izin untuk mengonfigurasi integrasi di AWS Security Agent Management Console

Important

Satu situs Atlassian hanya dapat dikaitkan dengan satu akun AWS per wilayah. Jika Anda perlu menghubungkan situs Bitbucket yang sama ke AWS Security Agent di akun AWS yang berbeda dalam wilayah yang sama, Anda harus terlebih dahulu menghapus integrasi yang ada.

Note

Harga aplikasi Atlassian Forge berlaku untuk integrasi ini. Untuk informasi selengkapnya, lihat Forge harga platform di dokumentasi Atlassian.

Daftarkan koneksi Bitbucket

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Pilih Tambahkan integrasi.
3. Pilih Bitbucket, lalu pilih Berikutnya.
4. Instal aplikasi AWS Security Agent Forge di ruang kerja Bitbucket Anda, mengikuti petunjuk di layar. Setelah instalasi, salin ID instalasi. Lihat the section called “Temukan ID instalasi Atlassian Anda”.
5. Di bidang ID Instalasi, tempel ID instalasi yang Anda salin dari Bitbucket.
6. Di bidang ruang kerja Bitbucket, masukkan slug ruang kerja Bitbucket Anda, misalnya. `acme-corp`
7. Pilih Izinkan.

Anda dialihkan ke Atlassian untuk memberi wewenang kepada AWS Security Agent untuk mengakses ruang kerja Bitbucket Anda. Setelah otorisasi selesai, Anda kembali ke konsol.

8. Di bagian Register details, masukkan nama Registrasi untuk koneksi ini. Karakter yang valid adalah huruf, angka, titik, garis bawah, dan tanda hubung.
9. Pilih Hubungkan.

> **Note**
>
> Penundaan penyediaan setelah menghubungkan
>
> Setelah Anda memilih Connect, penyediaan di sisi Atlassian dapat memakan waktu beberapa menit. Selama waktu ini, integrasi melaporkan status yang tertunda. Jika Anda mencoba memilih repositori atau mengaktifkan peninjauan kode sebelum penyediaan selesai, Anda mungkin melihat pesan bahwa penginstalan belum tersedia. Tunggu beberapa menit dan coba lagi.

Memecahkan masalah integrasi Bitbucket

Jika Anda mengalami masalah selama proses integrasi Bitbucket, gunakan panduan berikut untuk menyelesaikan masalah umum.

Tidak dapat menyelesaikan pendaftaran

Jika proses pendaftaran terganggu (browser ditutup, batas waktu sesi), aplikasi Forge dapat diinstal di situs Atlassian Anda tetapi tidak terdaftar di AWS Console.

Resolusi

- Kembali ke konsol AWS Security Agent dan mulai ulang proses integrasi.
- Aplikasi Forge tetap diinstal dan tidak perlu diinstal ulang.

Situs sudah terhubung ke akun AWS lain

Gejala

- Kesalahan yang menunjukkan situs Atlassian sudah dikaitkan dengan akun lain

Resolusi

- Satu situs Atlassian hanya dapat dihubungkan ke satu akun AWS per wilayah.
- Identifikasi akun AWS mana yang memiliki integrasi yang ada dan gunakan akun itu, atau hapus integrasi yang ada terlebih dahulu.

Instalasi tidak tersedia

Gejala

- Ketika Anda memilih repositori atau mengaktifkan peninjauan kode, Anda melihat pesan bahwa instalasi Bitbucket dalam status`PENDING_INSTALLATION`, bukan. `AVAILABLE`

Resolusi

- Instalasi masih disediakan di sisi Atlassian. Penyediaan biasanya selesai dalam beberapa menit. Tunggu beberapa menit, lalu coba lagi.
- Jika status tidak tersedia setelah beberapa menit, hapus integrasi dan daftarkan lagi. Sebelum mendaftar ulang, hapus instalan aplikasi Forge dari ruang kerja Bitbucket Anda. Untuk petunjuk, lihat Menghapus integrasi Bitbucket. Jika Anda mendaftar ulang tanpa mencopot pemasangan aplikasi Forge sebelumnya, penginstalan dapat macet dalam keadaan tertunda.

Langkah selanjutnya

Setelah menghubungkan Bitbucket ke AWS Security Agent:

- Arahkan ke Ruang Agen tempat Anda ingin menggunakan repositori ini
- Pilih Aktifkan peninjauan kode atau Setup pengujian penetrasi untuk menghubungkan repositori tertentu ke Ruang Agen Anda
- Aktifkan remediasi Kode untuk memungkinkan AWS Security Agent mengirimkan permintaan tarik dengan perbaikan kerentanan

Hapus integrasi Bitbucket

Hapus integrasi Bitbucket saat Anda tidak lagi memerlukan AWS Security Agent untuk mengakses repositori dari ruang kerja Bitbucket tertentu.

Prasyarat untuk penghapusan

Sebelum menghapus integrasi Bitbucket:

- Periksa Ruang Agen mana yang memiliki repositori yang terhubung dari integrasi ini
- Memahami dampaknya: Menghapus akan merusak tinjauan kode, konteks pengujian penetrasi, pemodelan ancaman, dan remediasi otomatis untuk semua repositori yang terhubung

Langkah 1: Hapus integrasi dari AWS Security Agent

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Temukan integrasi Bitbucket yang ingin Anda hapus.
3. Pilih integrasi.
4. Pilih Hapus.
5. Tinjau dialog konfirmasi dan pilih Konfirmasi penghapusan.

Langkah 2: Copot pemasangan aplikasi Forge

Setelah menghapus integrasi dari AWS Security Agent, hapus instalan aplikasi Forge dari situs Atlassian Anda:

1. Di Bitbucket, navigasikan ke pengaturan Workspace.
2. Pilih Forge Apps.
3. Temukan Connect dengan AWS Security Agent.
4. Pilih Hapus Instalasi.

> **Important**
>
> Aplikasi Forge tidak dihapus secara otomatis
>
> Menghapus integrasi di konsol AWS Security Agent tidak menghapus aplikasi Forge dari situs Atlassian Anda. Jika Anda berencana untuk mendaftarkan ruang kerja Bitbucket yang sama lagi, Anda harus menghapus aplikasi Forge terlebih dahulu. Jika tidak, instalasi baru bisa macet dalam keadaan tertunda.

Connect AWS Security Agent ke Confluence

Connect AWS Security Agent Anda ke Confluence Cloud untuk menyediakan konteks dokumentasi untuk penilaian keamanan. Tidak seperti penyedia kode, Confluence berfungsi sebagai sumber dokumentasi yang menyediakan model ancaman, dokumen arsitektur, spesifikasi API, dan materi lain yang meningkatkan kualitas tinjauan keamanan. Sebelum memulai, tinjau the section called "Bagaimana integrasi bekerja dengan Agen Spaces" untuk memahami bagaimana pendaftaran digunakan kembali di seluruh Ruang Agen.

Integrasi pertemuan melayani berbagai tujuan:

- Konteks tinjauan desain - Menyediakan dokumen arsitektur dan spesifikasi desain untuk ulasan desain keamanan
- Pemodelan ancaman - Menyediakan model ancaman dan dokumentasi sistem yang ada untuk analisis ancaman
- Konteks pengujian penetrasi - Menyediakan dokumentasi aplikasi untuk pemahaman yang lebih dalam selama pengujian penetrasi

Menghubungkan Confluence ke AWS Security Agent memerlukan penginstalan aplikasi AWS Security Agent Forge di situs Atlassian Anda dan menyelesaikan alur otorisasi OAuth.

> **Note**
>
> AWS Security Agent hanya mendukung Confluence Cloud. Pusat Data Confluence dan Server Confluence tidak didukung.

Bagaimana integrasi Confluence bekerja

Confluence adalah penyedia dokumentasi daripada penyedia kode sumber. Setelah Anda menginstal aplikasi Forge dan menghubungkan spasi atau halaman di AWS Management Console, AWS Security Agent dapat mengakses konten Confluence Anda untuk memberikan konteks selama penilaian keamanan.

AWS Security Agent membaca konten halaman untuk memahami arsitektur aplikasi, persyaratan keamanan, dan keputusan desain Anda. Konteks ini meningkatkan kualitas dan relevansi temuan keamanan selama tinjauan desain, tinjauan kode, dan tes penetrasi.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Situs Confluence Cloud dengan akses admin
- Akun Atlassian dengan hak administrator situs
- Izin untuk mengonfigurasi integrasi di AWS Security Agent Management Console

> **Important**
>
> Satu situs Atlassian hanya dapat dikaitkan dengan satu akun AWS per wilayah. Jika Anda perlu menghubungkan situs Confluence yang sama ke AWS Security Agent di akun AWS yang berbeda dalam wilayah yang sama, Anda harus terlebih dahulu menghapus integrasi yang ada.

> **Note**
>
> Harga aplikasi Atlassian Forge berlaku untuk integrasi ini. Untuk informasi selengkapnya, lihat Forge harga platform di dokumentasi Atlassian.

Daftarkan koneksi Confluence

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Pilih Tambahkan integrasi.
3. Pilih Confluence, lalu pilih Next.
4. Instal aplikasi AWS Security Agent Forge di situs Atlassian Anda, mengikuti petunjuk di layar. Setelah instalasi, salin ID instalasi. Lihat the section called “Temukan ID instalasi Atlassian Anda”.
5. Di bidang URL situs Confluence, masukkan URL situs Anda, misalnya. `https://acme.atlassian.net`
6. Di bidang ID Instalasi, tempel ID instalasi yang Anda salin dari Confluence.
7. Pilih Izinkan.

   Anda dialihkan ke Atlassian untuk memberi wewenang kepada AWS Security Agent untuk mengakses situs Confluence Anda. Setelah otorisasi selesai, Anda kembali ke konsol.

8. Di bagian Register details, masukkan nama Registrasi untuk koneksi ini. Karakter yang valid adalah huruf, angka, titik, garis bawah, dan tanda hubung.
9. Pilih Hubungkan.

Pilih halaman untuk Ruang Agen

Setelah Anda mendaftarkan integrasi Confluence, hubungkan halaman tertentu ke Ruang Agen. Memilih halaman memberikan AWS Security Agent read (fetch) akses ke konten halaman tersebut. Tidak ada opsi kemampuan per halaman — agen membaca setiap halaman yang terhubung. Pada langkah peninjauan wizard sambungkan, Anda dapat menghapus halaman apa pun yang tidak ingin diakses agen.

Memecahkan masalah integrasi Confluence

Jika Anda mengalami masalah selama proses integrasi Confluence, gunakan panduan berikut untuk menyelesaikan masalah umum.

Tidak dapat menyelesaikan pendaftaran

Jika proses pendaftaran terganggu, aplikasi Forge dapat diinstal di situs Atlassian Anda tetapi tidak terdaftar di AWS Console.

Resolusi

- Kembali ke konsol AWS Security Agent dan mulai ulang proses integrasi.
- Aplikasi Forge tetap diinstal dan tidak perlu diinstal ulang.

Aplikasi Forge dihapus dari Atlassian

Jika aplikasi AWS Security Agent Forge dihapus dari situs Atlassian Anda saat integrasi masih ada di AWS Security Agent:

Gejala

- Integrasi muncul di AWS Console tetapi tidak dapat mengakses konten Confluence
- Kesalahan saat mencoba membuat daftar atau membaca halaman

Resolusi

- Instal ulang aplikasi Forge dari Atlassian Marketplace
- Jika masalah berlanjut, hapus integrasi di AWS Console dan daftarkan ulang

Situs sudah terhubung ke akun AWS lain

Resolusi

- Satu situs Atlassian hanya dapat dihubungkan ke satu akun AWS per wilayah.
- Identifikasi akun AWS mana yang memiliki integrasi yang ada dan gunakan akun itu, atau hapus integrasi yang ada terlebih dahulu.

Langkah selanjutnya

Setelah menghubungkan Confluence ke AWS Security Agent:

- Arahkan ke Ruang Agen tempat Anda ingin menggunakan dokumentasi ini
- Pilih halaman tertentu untuk disertakan sebagai konteks untuk tinjauan desain dan tes penetrasi
- Unggah dokumentasi tambahan melalui S3 jika diperlukan (lihatthe section called “Menyediakan sumber daya agen dari bucket S3”)

Hapus integrasi Confluence

Hapus integrasi Confluence saat Anda tidak lagi memerlukan AWS Security Agent untuk mengakses dokumentasi dari situs Confluence tertentu.

Prasyarat untuk penghapusan

Sebelum menghapus integrasi Confluence:

- Periksa Ruang Agen mana yang memiliki halaman yang terhubung dari integrasi ini
- Memahami dampaknya: Menghapus akan merusak konteks dokumentasi untuk tinjauan desain, pengujian penetrasi, dan pemodelan ancaman di semua Ruang Agen menggunakan konten dari integrasi ini

Langkah 1: Hapus integrasi dari AWS Security Agent

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Temukan integrasi Confluence yang ingin Anda hapus.
3. Pilih integrasi.
4. Pilih Hapus.
5. Tinjau dialog konfirmasi dan pilih Konfirmasi penghapusan.

Langkah 2: Copot pemasangan aplikasi Forge (opsional)

Setelah menghapus integrasi dari AWS Security Agent, Anda dapat secara opsional menghapus aplikasi Forge dari situs Atlassian Anda:

1. Di Confluence, arahkan ke administrasi Confluence.
2. Pilih Aplikasi.
3. Temukan Connect dengan AWS Security Agent.
4. Pilih Hapus Instalasi.

Connect ke kontrol sumber yang dihosting secara pribadi

> **Important**
>
> Koneksi pribadi dalam rilis pratinjau untuk AWS Security Agent dan dapat berubah sewaktu-waktu.

AWS Security Agent dapat terhubung ke sistem kontrol sumber yang berjalan di jaringan pribadi, seperti GitLab Self-Managed instans dan Server GitHub Perusahaan. Koneksi pribadi menggunakan Amazon VPC Lattice untuk membangun konektivitas aman antara AWS Security Agent dan infrastruktur pribadi Anda tanpa mengekspos sistem Anda ke internet publik.

Cara kerja koneksi pribadi

Koneksi pribadi menciptakan jalur jaringan yang aman antara AWS Security Agent dan sumber daya target di VPC Anda. Di bawah tenda, AWS Security Agent menggunakan Amazon VPC Lattice untuk membangun konektivitas ini. VPC Lattice adalah layanan jaringan aplikasi AWS untuk

menghubungkan, mengamankan, dan memantau lalu lintas antara layanan di seluruh VPC dan akun, tanpa Anda mengelola infrastruktur jaringan yang mendasarinya.

Saat Anda membuat koneksi pribadi:

1. Anda menyediakan VPC, subnet, dan grup keamanan (opsional) yang memiliki konektivitas jaringan ke layanan target Anda.
2. AWS Security Agent membuat gateway sumber daya yang dikelola layanan dan menyediakan antarmuka jaringan elastis (ENI) dalam subnet yang Anda tentukan.
3. Agen menggunakan gateway sumber daya untuk mengarahkan lalu lintas ke alamat IP atau nama DNS layanan target Anda melalui jalur jaringan pribadi.

Service-managed vs. koneksi pribadi yang dikelola sendiri

AWS Security Agent mendukung dua mode untuk koneksi pribadi:

Service-managed(direkomendasikan untuk pengaturan sederhana):

- AWS Security Agent membuat dan mengelola gateway sumber daya VPC Lattice untuk Anda.
- Layanan target harus berjalan di akun AWS yang sama di mana Ruang Agen dibuat.
- Tidak dapat menjangkau beberapa akun AWS.

Self-managed(untuk pengaturan lanjutan atau lintas akun):

- Anda membuat dan mengelola konfigurasi sumber daya VPC Lattice Anda sendiri.
- Anda memberikan Nama Sumber Daya Amazon (ARN) dari konfigurasi sumber daya yang ada.
- Mendukung akses lintas akun (pelanggan mengelola jaringan lintas akun).

Prasyarat

Sebelum membuat koneksi pribadi, verifikasi bahwa Anda memiliki:

- Ruang Agen yang aktif
- Layanan target yang dapat dijangkau secara pribadi (GitLab Self-Managed atau Server GitHub Perusahaan) di alamat IP pribadi atau nama DNS yang dikenal
- Layanan target harus melayani lalu lintas HTTPS dengan versi TLS minimum 1.2

- Subnet di VPC Anda (1-20 subnet). Sebaiknya pilih subnet di beberapa Availability Zone untuk ketersediaan tinggi.
- (Opsional) Grup keamanan untuk mengontrol lalu lintas ke ENI (hingga 5)

> **Note**
>
> Availability Zone berikut tidak didukung oleh VPC Lattice:`use1-az3`,,,`usw1-az2`,,`apne1-az3`, `apne2-az2euc1-az2`,`euw1-az4`. `cac1-az3` `ilc1-az2`

> **Note**
>
> Koneksi pribadi adalah sumber daya tingkat akun. Setelah membuat koneksi pribadi, Anda dapat menggunakannya kembali di beberapa integrasi dan Ruang Agen yang perlu menjangkau host yang sama.

> **Note**
>
> Saat Anda memilih koneksi pribadi untuk penyedia yang menggunakan otentikasi OAuth, koneksi pribadi berlaku untuk titik akhir penyedia dan titik akhir pertukaran token. Pastikan koneksi pribadi dikonfigurasi dengan alamat host yang dapat merutekan lalu lintas ke kedua titik akhir.

Buat koneksi pribadi

1. Di AWS Security Agent Management Console, navigasikan ke Integrasi.
2. Pilih tab Sambungan pribadi.
3. Pilih Buat koneksi pribadi.
4. Di bawah Detail koneksi, masukkan Nama. Nama dapat berisi huruf kecil, angka, dan tanda hubung, dan harus dimulai dan diakhiri dengan huruf atau angka.
5. Di bagian Detail koneksi, pilih cara AWS Security Agent terhubung ke layanan target Anda:

- Agar AWS Security Agent membuat dan mengelola koneksi, biarkan Gunakan konfigurasi sumber daya yang ada dibersihkan, lalu lengkapi bagian Lokasi Sumber Daya, Kontrol akses, dan detail target Layanan.
- Untuk merutekan melalui konfigurasi sumber daya VPC Lattice yang telah Anda buat, pilih Gunakan konfigurasi sumber daya yang ada, lalu selesaikan bagian Konfigurasi sumber daya yang ada. Bagian yang dikelola layanan tidak berlaku.

6. (Service-managed) Di bawah Lokasi sumber daya:
   a. Pilih VPC tempat sumber daya Anda berada.
   b. Pilih satu atau lebih Subnet. Sebaiknya pilih subnet di setidaknya dua Availability Zone.
   c. (Opsional) Pilih jenis alamat IP (IPv4, IPv6, atau Dual stack).
7. (Service-managed) Di bawah kontrol Akses:
   a. Pilih Grup keamanan yang terkait dengan sumber daya yang terhubung.
   b. (Opsional) Di bawah Konfigurasi lanjutan, masukkan rentang Port — hingga 11 port atau rentang TCP yang dipisahkan koma, misalnya. `443, 8080-8090`
8. (Service-managed) Di bawah rincian target Layanan:
   a. Di bidang alamat Host, masukkan nama host DNS atau alamat IP dari layanan target Anda.
   b. Untuk resolusi DNS, pilih Publik untuk alamat host yang dapat diselesaikan secara publik, atau Dalam VPC (DNS pribadi) untuk alamat yang dapat diselesaikan hanya di dalam VPC.
   c. (Opsional) Di bidang kunci publik Sertifikat, masukkan kunci PEM-encoded publik jika target Anda menggunakan sertifikat yang ditandatangani sendiri atau otoritas sertifikat pribadi. Ini memungkinkan AWS Security Agent mempercayai koneksi TLS.
9. (Self-managed) Di bawah Konfigurasi sumber daya yang ada, untuk konfigurasi Sumber Daya, pilih konfigurasi sumber daya VPC Lattice dari daftar, atau masukkan ID konfigurasi sumber daya (dimulai dengan`rcfg-`) atau ARN-nya. Daftar ini menunjukkan konfigurasi sumber daya di Wilayah saat ini.
10. Pilih Buat koneksi.

    Status koneksi berubah menjadi Create in progress. Ini bisa memakan waktu hingga 10 menit. Setelah selesai, status berubah menjadi Tersedia.

Gunakan koneksi pribadi dengan integrasi

Saat Anda mendaftarkan integrasi Server GitLab Self-Managed atau GitHub Enterprise, pilih Connect to endpoint menggunakan koneksi pribadi, lalu pilih koneksi Anda dari daftar Sambungan pribadi.

AWS Security Agent merutekan lalu lintas ke penyedia melalui koneksi tersebut. Hanya koneksi dengan status Tersedia yang muncul dalam daftar.

Anda juga dapat memilih Buat koneksi pribadi saat pendaftaran. AWS Security Agent mempertahankan draf pendaftaran Anda saat Anda membuat koneksi, lalu mengembalikan Anda ke alur pendaftaran.

Keamanan

- Tidak ada eksposur internet publik - Semua lalu lintas tetap berada di jaringan AWS.
- Customer-controlled grup keamanan - Anda mengelola peraturan lalu lintas masuk dan keluar.
- Service-linked peran dengan hak istimewa paling sedikit - AWS Security Agent menggunakan peran terkait layanan yang dicakup oleh sumber daya yang ditandai. `AWSSecurityAgentManaged`

> **Note**
>
> Jika organisasi Anda memiliki kebijakan kontrol layanan (SCP) yang membatasi tindakan VPC Lattice API, gateway sumber daya yang dikelola layanan dibuat melalui peran terkait layanan. Pastikan SCP Anda mengizinkan tindakan yang diperlukan untuk peran terkait layanan.

Verifikasi koneksi pribadi

Setelah koneksi pribadi mencapai status Tersedia, verifikasi konektivitas:

- Pastikan layanan target Anda berjalan dan menerima koneksi pada port yang diharapkan
- Verifikasi bahwa grup keamanan yang dilampirkan ke ENI memungkinkan lalu lintas keluar pada port target
- Verifikasi bahwa grup keamanan layanan Anda mengizinkan lalu lintas masuk dari IP pesawat data VPC Lattice dalam rentang CIDR VPC Anda
- Konfirmasikan bahwa tabel rute subnet memungkinkan lalu lintas antara subnet ENI dan layanan Anda

Hapus koneksi pribadi

1. Di konsol AWS Security Agent, navigasikan ke Integrasi.
2. Pilih tab Sambungan pribadi.
3. Pilih koneksi pribadi yang ingin Anda hapus.
4. Pilih Hapus.

> **Important**
>
> Anda tidak dapat menghapus koneksi pribadi yang saat ini digunakan oleh integrasi. Hapus integrasi terlebih dahulu, lalu hapus koneksi pribadi.

Langkah selanjutnya

- the section called “Connect AWS Security Agent ke GitLab Self-Managed”- Connect GitLab instance pribadi
- the section called “Connect AWS Security Agent ke GitHub Enterprise Server”- Connect Server GitHub Enterprise pribadi

Connect agen ke sumber daya VPC pribadi

Jika aplikasi yang ingin Anda jalankan uji penetrasi tidak tersedia di internet publik, Anda perlu menyediakan AWS Security Agent dengan konfigurasi VPC. AWS Security Agent akan menggunakan konfigurasi VPC ini, termasuk VPC, subnet, dan grup keamanan, untuk mengakses aplikasi.

> **Note**
>
> Saat menguji titik akhir dalam VPC pribadi, hanya titik akhir yang menyelesaikan IP dalam rentang IP pribadi yang diketahui yang diizinkan (lihat blok CIDR VPC untuk informasi lebih lanjut). Rentang IPv4 dan IPv6 berikut diperbolehkan:
>
> ```
> 10.0.0.0/8
> 172.16.0.0/12
> 192.168.0.0/16
> fd00::/8
> ```

> **Note**
>
> Saat menghubungkan ke subnet, AWS Security Agent akan membuat ENI (Elastic Network Interface) di subnet yang dikonfigurasi untuk pengujian penetrasi. ENI ini tidak memiliki alamat IP publik terkait, yang berarti tidak dapat berkomunikasi dengan VPC Internet Gateways di subnet publik. Jika tes penetrasi Anda memerlukan akses internet terbuka, silakan gunakan subnet pribadi dengan VPC NAT Gateway terkait sebagai gantinya

Anda memberikan AWS Security Agent akses umum ke VPC dari AWS Management Console. Di aplikasi web Agen Keamanan, pengguna memilih konfigurasi spesifik untuk pengujian penetrasi.

Untuk menambahkan VPC di Ruang Agen

1. Arahkan ke halaman ikhtisar Ruang Agen
2. Pilih Tindakan dan kemudian Edit konfigurasi uji penetrasi
3. Di bawah judul VPC, tentukan grup VPC, Subnet, dan Keamanan

Anda dapat menambahkan hingga 5 VPC.

Izin peran layanan ruang agen yang diperlukan

Untuk menjalankan pengujian penetrasi dengan VPC, peran layanan ruang agen Anda harus menyertakan izin berikut:

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

Untuk memilih konfigurasi VPC tertentu untuk pengujian penetrasi di aplikasi web Agen Keamanan

1. Arahkan ke halaman ikhtisar Tes Penetrasi
2. Pilih tes penetrasi yang Anda perlukan untuk menambahkan konfigurasi VPC, lalu pilih Ubah detail pentest
3. Pilih Berikutnya untuk mencapai bagian Sumber Daya VPC
4. Pilih grup VPC, Subnet, dan Keamanan
5. Pilih Berikutnya untuk mencapai bagian terakhir dan Simpan tes penetrasi

> **Note**
>
> Cross-account pengujian penetrasi saat ini didukung untuk sumber daya VPC (subnet dan grup keamanan) yang dibagikan menggunakan AWS Resource Access Manager. Rahasia Secrets Manager dan fungsi Lambda yang digunakan untuk kredensi autentikasi harus dikonfigurasi di akun AWS yang sama dengan penyiapan AWS Security Agent Anda.

Menjalankan uji penetrasi terhadap sumber daya VPC di akun AWS lain

Anda dapat menjalankan pengujian penetrasi terhadap sumber daya VPC yang dibagikan dengan akun Anda menggunakan AWS Resource Access Manager. Kedua akun harus menjadi bagian dari AWS Organization yang sama.

1. (Opsional) Aktifkan berbagi sumber daya otomatis untuk organisasi AWS Anda

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

2. Menggunakan kredensi dari akun AWS yang memiliki sumber daya VPC, berbagi subnet dan sumber daya grup keamanan dengan akun pemilik pengujian penetrasi

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

3. Arahkan ke halaman ikhtisar Ruang Agen
4. Pilih Tes Penetrasi dan temukan nama peran Layanan
5. Verifikasi bahwa peran IAM memberikan akses ke sumber daya VPC bersama
6. Pilih Tindakan dan kemudian Edit konfigurasi uji penetrasi
7. Di bawah judul VPC, tentukan grup VPC, Subnet, dan Keamanan bersama dan simpan konfigurasi yang diperbarui.
8. Arahkan ke halaman ikhtisar Tes Penetrasi di aplikasi web AWS Security Agent
9. Pilih tes penetrasi yang Anda perlukan untuk menambahkan konfigurasi VPC, lalu pilih Ubah detail pentest
10. Perbarui tes penetrasi untuk menggunakan sumber daya VPC bersama

Konfigurasikan kemampuan

Aktifkan dan konfigurasikan pengujian penetrasi, peninjauan kode, dan pemodelan ancaman untuk Ruang Agen Anda, termasuk remediasi, sumber S3, dan domain target.

Aktifkan uji penetrasi

Konfigurasikan AWS Security Agent untuk menjalankan pengujian penetrasi otonom pada aplikasi Anda. Penyiapan ini memungkinkan AWS Security Agent untuk mengakses sumber daya AWS Anda, memverifikasi kepemilikan domain, dan melakukan pengujian keamanan komprehensif yang mengidentifikasi kerentanan yang dapat dieksploitasi dalam aplikasi web dan API Anda.

Mengaktifkan pengujian penetrasi memungkinkan AWS Security Agent untuk terus memvalidasi keamanan aplikasi Anda tanpa penundaan pengujian manual. Dengan mengonfigurasi integrasi AWS yang diperlukan, Anda memastikan AWS Security Agent dapat menguji aplikasi publik dan pribadi, mencatat temuan CloudWatch, dan mengakses kredensyal untuk pengujian yang diautentikasi.

Dalam prosedur ini, Anda akan mengonfigurasi domain target, secara opsional mengatur VPC, CloudWatch logging, penyimpanan kredensyal, dan fungsi Lambda untuk pengujian, mengonfigurasi integrasi S3 untuk menyediakan konteks tambahan, dan mengatur akses layanan melalui peran IAM.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Akun AWS dengan izin administratif untuk membuat peran IAM
- Kemampuan verifikasi kepemilikan domain (modifikasi catatan DNS atau HTTP)
- Targetkan domain atau aplikasi untuk diuji
- (Opsional) Detail konfigurasi VPC jika menguji aplikasi pribadi
- (Opsional) bucket S3 jika menyediakan artefak tambahan ke AWS Security Agent

Langkah 1: Konfigurasikan domain

Pada langkah pertama wizard, tentukan domain target yang ingin Anda uji dan bagaimana AWS Security Agent harus memverifikasi kepemilikan.

1. Di bagian Domain target, masukkan domain Anda di bidang Domain.
2. Pilih metode Verifikasi:
   - DNS_TXT — Buktikan kepemilikan domain dengan menambahkan catatan TXT ke konfigurasi DNS domain Anda.
   - HTTP_ROUTE - Buktikan kepemilikan domain dengan menghosting file verifikasi di URL tertentu di domain Anda.

- PRIVATE_VPC - Hanya dapat digunakan untuk pengujian penetrasi VPC pribadi. Memverifikasi bahwa domain menyelesaikan ke IP dalam rentang CIDR pribadi
- Untuk informasi selengkapnya, lihat the section called “Aktifkan domain aplikasi untuk pengujian penetrasi”.

3. Pilih Tambahkan domain lain untuk menambahkan domain tambahan (total hingga 5).
4. Pilih Berikutnya untuk melanjutkan ke verifikasi domain.

Langkah 2: Verifikasi domain

Pada langkah kedua wizard, verifikasi kepemilikan setiap domain yang Anda konfigurasikan. AWS Security Agent memerlukan kepemilikan terverifikasi sebelum dapat melakukan pengujian penetrasi.

1. Tinjau domain yang tercantum dalam tabel domain Target.
2. Untuk setiap domain, pilih dan picu verifikasi berdasarkan metode yang Anda pilih:
   - Route 53 domain (akun AWS yang sama): Pilih One-click verifikasi. AWS Security Agent secara otomatis membuat catatan DNS dan menyelesaikan verifikasi.
   - DNS TXT (penyedia DNS lainnya): Salin token verifikasi, tambahkan catatan TXT dengan registrar DNS Anda, lalu pilih domain dan pilih Verifikasi.
   - Rute HTTP: Tempatkan token verifikasi di jalur rute yang diperlukan di server web Anda, lalu pilih domain dan pilih Verifikasi. Lihat perinciannya di the section called “Aktifkan domain aplikasi untuk pengujian penetrasi”.
3. Pilih Berikutnya untuk melanjutkan ke konfigurasi opsional.

> **Note**
>
> Sub-domains domain terverifikasi tidak memerlukan verifikasi individu saat menggunakan DNS TXT atau verifikasi rute HTTP. Untuk verifikasi VPC Pribadi, nama domain titik akhir target harus sesuai dengan nama domain lengkap yang dikonfigurasi untuk verifikasi.

> **Note**
>
> Untuk domain pribadi di dalam VPC, Anda dapat melanjutkan meskipun status verifikasi domain UNREACHABLE. AWS Security Agent akan mencoba verifikasi domain untuk titik akhir pribadi pada awal setiap pentest run.

Langkah 3: (Opsional) Konfigurasikan kemampuan tambahan

Langkah ketiga wizard memungkinkan Anda mengonfigurasi sumber daya AWS opsional untuk memperluas cakupan pengujian penetrasi Anda. Semua bagian dalam langkah ini adalah opsional dan diciutkan secara default.

(Opsional) Konfigurasikan pengaturan VPC

Jika Anda berencana untuk menguji domain target pribadi yang dihosting dalam VPC, konfigurasikan pengaturan VPC untuk AWS Security Agent. Bagian ini opsional dan diciutkan secara default.

1. Perluas bagian VPC.
2. Di dropdown VPC, pilih VPC yang menghosting domain target pribadi Anda.
3. Di dropdown Subnet, pilih subnet VPC yang harus digunakan AWS Security Agent:

> **Tip**
>
> Untuk ketersediaan tinggi, pilih beberapa subnet dari beberapa Availability Zone. Pastikan subnet Anda menyertakan gateway NAT untuk konektivitas keluar.

4. Di menu tarik-turun grup Keamanan, pilih grup keamanan VPC yang harus digunakan AWS Security Agent:

> **Important**
>
> Pastikan grup keamanan Anda mengizinkan koneksi keluar untuk AWS Security Agent untuk melakukan pengujian penetrasi.

(Opsional) Konfigurasikan CloudWatch logging

Konfigurasikan CloudWatch untuk menyimpan log dari pengujian penetrasi Anda berjalan. Bagian ini opsional dan diciutkan secara default.

1. Perluas bagian CloudWatch log.
2. Di dropdown Grup Log, pilih grup log yang ada CloudWatch
3. Jika tidak memilih grup log apa pun, AWS Security Agent akan membuat grup log bernama `/aws/securityagent/<agent name>/<pentest id>` dengan izin yang sesuai.

> **Note**
>
> Pastikan peran IAM Anda memiliki izin untuk menulis ke grup CloudWatch log yang dipilih.

(Opsional) Konfigurasikan Rahasia untuk kredensyal pengujian

Jika aplikasi Anda memerlukan kredensyal autentikasi untuk pengujian, AWS Security Agent dapat mengambilnya dengan aman dari AWS Secrets Manager. Bagian ini opsional dan diciutkan secara default.

1. Perluas bagian Rahasia.
2. Pilih rahasia AWS Secrets Manager yang berisi kredensyal untuk aplikasi Anda.
3. Saat mengonfigurasi tes penetrasi di aplikasi web, Anda dapat mereferensikan rahasia ini untuk pengujian yang diautentikasi.

> **Important**
>
> Kredensyal dienkripsi dan disimpan di AWS Secrets Manager. Pastikan peran IAM Anda memiliki izin untuk mengakses Secrets Manager for AWS Security Agent untuk menggunakan kredensyal ini selama pengujian.

(Opsional) Konfigurasikan fungsi Lambda untuk kredensyal pengujian

Konfigurasikan fungsi Lambda yang dapat memberikan kredensyal untuk aplikasi Anda selama pengujian. Bagian ini opsional dan diciutkan secara default.

1. Perluas bagian fungsi Lambda.
2. Pilih fungsi Lambda yang dapat memberikan kredensyal otentikasi untuk aplikasi Anda.
3. AWS Security Agent akan menjalankan fungsi-fungsi ini selama pengujian penetrasi untuk mendapatkan kredensyal secara dinamis.

> **Note**
>
> Pastikan peran IAM Anda memiliki izin untuk menjalankan fungsi Lambda yang ditentukan. Fungsi Lambda harus mengembalikan kredensyal dalam format yang diharapkan untuk AWS Security Agent untuk digunakan selama pengujian.

(Opsional) Konfigurasikan ember S3

Berikan detail bucket S3 jika Anda berencana mengunggah dokumen atau artefak untuk diberikan sebagai masukan ke AWS Security Agent. Bagian ini opsional dan diciutkan secara default.

1. Perluas bagian bucket S3.
2. Di bidang Bucket, masukkan atau cari nama bucket S3 Anda.

> **Note**
>
> Anda juga dapat menghubungkan GitHub repositori nanti atau mengunggah file langsung di aplikasi web. Informasi yang diberikan dapat memastikan cakupan menyeluruh, mengurangi kesalahan positif, dan memberikan hasil yang dapat ditindaklanjuti.

(Opsional) Konfigurasikan akses layanan

AWS Security Agent memerlukan peran IAM untuk mengakses sumber daya AWS Anda (VPC CloudWatch , grup log, Secrets Manager, fungsi Lambda, dll.) untuk pengujian penetrasi. Bagian ini opsional dan diciutkan secara default.

1. Perluas bagian Akses layanan.
2. Secara default, AWS Security Agent menggunakan peran IAM default dengan izin yang diperlukan untuk pengujian penetrasi.

3. Untuk menyesuaikan peran IAM, pilih salah satu opsi berikut:
   a. Buat peran default — AWS Security Agent secara otomatis membuat peran IAM baru dengan izin yang diperlukan
   b. Menggunakan peran layanan yang ada — Pilih peran IAM yang ada dari menu tarik-turun
4. Jika menggunakan peran yang ada:
   a. Pilih menu tarik-turun di bawah Pilih peran yang ada
   b. Pilih peran IAM Anda dari daftar
   c. Pilih ikon penyegaran untuk memperbarui daftar jika diperlukan

> **Note**
>
> Peran IAM default mencakup izin untuk mengakses sumber daya VPC, grup log, Secrets Manager CloudWatch , fungsi Lambda, dan layanan lain yang diperlukan untuk pengujian penetrasi. Disarankan untuk menggunakan peran IAM default kecuali Anda memiliki persyaratan keamanan khusus.

Langkah 4: Simpan dan aktifkan pengujian penetrasi

Setelah mengonfigurasi semua pengaturan yang diperlukan, aktifkan pengujian penetrasi untuk agen AWS Security Agent Anda.

1. Tinjau semua bagian konfigurasi untuk memastikan akurasi.
2. Pilih Simpan.
3. AWS Security Agent memvalidasi konfigurasi Anda dan membuat sumber daya AWS yang diperlukan.

Langkah selanjutnya

Setelah mengaktifkan pengujian penetrasi:

- Konfigurasikan cakupan pengujian penetrasi di aplikasi web AWS Security Agent
- Siapkan preferensi notifikasi untuk temuan pengujian
- Tinjau dan tanggapi temuan uji penetrasi saat ditemukan
- (Opsional) Konfigurasikan repositori tambahan untuk remediasi temuan

Untuk informasi selengkapnya tentang menjalankan dan mengelola pengujian penetrasi, lihat dokumentasi aplikasi web AWS Security Agent.

Aktifkan tinjauan kode permintaan tarik untuk GitHub repositori

Tinjauan kode permintaan tarik sekarang dikonfigurasi sebagai bagian dari panduan penyiapan peninjauan kode. Saat Anda menghubungkan GitHub repositori ke Ruang Agen, Anda dapat mengaktifkan komentar permintaan tarik untuk masing-masing repositori agar AWS Security Agent secara otomatis menganalisis permintaan tarik dan memposting temuan keamanan.

Untuk petunjuk penyiapan lengkap, lihatthe section called “Aktifkan peninjauan kode”.

Untuk informasi tentang bagaimana temuan permintaan tarik muncul GitHub dan bagaimana menanggapinya, lihatthe section called “Tinjau temuan keamanan kode dalam permintaan tarik”.

Aktifkan peninjauan kode

Konfigurasikan Ruang Agen Anda untuk mengaktifkan peninjauan kode dengan menghubungkan kode sumber dari GitHub repositori atau bucket S3. Peninjauan kode menganalisis kode sumber Anda untuk kerentanan keamanan dan kepatuhan terhadap persyaratan keamanan khusus organisasi Anda.

Menyiapkan konfigurasi peninjauan kode adalah Space-wide operasi Agen. Integrasi dan bucket S3 yang Anda sambungkan dibagi di seluruh kemampuan, termasuk peninjauan kode dan pengujian penetrasi.

Setelah menyelesaikan penyiapan, pengguna dapat membuat dan menjalankan tinjauan kode di aplikasi web AWS Security Agent untuk memindai repositori dan sumber S3 untuk masalah keamanan.

> **Note**
>
> Jika Anda sudah memiliki GitHub repositori yang terhubung ke Ruang Agen Anda (misalnya, melalui pengaturan pengujian penetrasi), peninjauan kode sudah diaktifkan. Anda dapat melewati pengaturan ini dan langsung menuju ke aplikasi web untuk membuat ulasan kode. Lihat the section called “Buat ulasan kode”.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Ruang Agen yang dibuat di AWS Management Console (lihatthe section called “Buat Ruang Agen”)
- Setidaknya satu dari masukan kode sumber berikut:
  - Akun GitHub organisasi atau pengguna dengan GitHub Aplikasi AWS Security Agent yang diinstal (lihatthe section called “Connect AWS Security Agent ke GitHub repositori”)
  - Bucket S3 yang berisi kode sumber yang ingin Anda tinjau
- Izin untuk mengonfigurasi integrasi untuk Ruang Agen Anda
- (Opsional) Setidaknya satu persyaratan keamanan diaktifkan dalam paket persyaratan keamanan jika Anda berencana untuk menggunakan validasi persyaratan keamanan (lihatthe section called “Kelola persyaratan keamanan”)

Mengakses wizard penyiapan peninjauan kode

Arahkan ke pengaturan peninjauan kode untuk Ruang Agen Anda.

1. Di konsol AWS Security Agent, pilih Ruang Agen Anda.
2. Pilih Aktifkan ulasan kode dari salah satu lokasi berikut:
   - Kartu peninjauan Kode
   - Tab Ulasan kode, lalu pilih Aktifkan tinjauan kode

> **Tip**
>
> Jika Anda ingin AWS Security Agent menangani umpan balik peninjauan kode secara otomatis, aktifkan perbaikan Kode juga. Lihat the section called “Memungkinkan pengguna untuk memulai remediasi uji penetrasi dan temuan tinjauan kode” untuk detail.

Anda diarahkan ke wizard konfigurasi peninjauan kode Pengaturan.

Langkah 1: Connect integrasi, repo, dan bucket

Pada langkah pertama wizard, sambungkan input kode sumber Anda dan konfigurasikan pengaturan tinjauan kode. Anda harus menambahkan setidaknya satu GitHub repositori atau ember S3 untuk melanjutkan.

> **Important**
>
> Integrasi dan bucket S3 yang dikonfigurasi di sini dibagikan di seluruh Ruang Agen Anda. Perubahan berlaku untuk peninjauan kode dan kemampuan pengujian penetrasi.

Connect GitHub repositori

Tambahkan GitHub repositori dari GitHub organisasi resmi atau akun pengguna Anda. Memilih Tambah membuka GitHub panduan Connect dua langkah, di mana Anda pertama kali memilih repositori dan kemudian mengonfigurasi tindakan apa yang dapat dilakukan AWS Security Agent pada masing-masing.

1. Di bagian Integrasi terhubung, pilih Tambah.
2. Pilih GitHub registrasi yang berisi repositori yang ingin Anda tinjau.
3. Pada langkah Connect GitHub repositories, pilih kotak centang untuk setiap repositori yang ingin Anda sambungkan.
4. Pilih Berikutnya untuk pergi ke Kelola kemampuan.

> **Note**
>
> Repositori yang terhubung diakses hanya-baca untuk analisis kode selama peninjauan kode dan pengujian penetrasi. Anda mengonfigurasi tindakan menulis seperti memposting komentar ulasan dan membuka permintaan tarik remediasi pada langkah berikutnya.

> **Note**
>
> Jika Anda belum mendaftarkan GitHub integrasi, pilih Pengaturan untuk menavigasi ke halaman Integrasi tempat Anda dapat mengotorisasi Aplikasi AWS Security Agent GitHub .

Untuk informasi selengkapnya, lihat the section called “Connect AWS Security Agent ke GitHub repositori”.

Konfigurasikan GitHub kemampuan repositori

Pada langkah Kelola kemampuan GitHub wizard Connect, pilih apa yang dapat dilakukan AWS Security Agent di setiap repositori. Komentar tinjauan kode dan remediasi Kode ditetapkan secara independen per repositori.

1. Untuk setiap repositori, alihkan komentar tinjauan Kode agar AWS Security Agent memposting temuan keamanan sebagai komentar pada permintaan tarik.
2. Untuk setiap repositori, aktifkan remediasi Kode agar AWS Security Agent memperbaiki temuan. Saat diaktifkan, pengguna aplikasi web dapat meminta permintaan tarik yang memperbaiki temuan, dan pembuat permintaan tarik dapat berkomentar `@AWS-Security-Agent fix all findings.` agar agen menangani komentar ulasannya.
3. Pilih Simpan untuk menerapkan pilihan Anda dan kembali ke panduan pengaturan peninjauan kode.

Ketika komentar tinjauan Kode diaktifkan untuk repositori:

- AWS Security Agent secara otomatis menganalisis permintaan tarik saat ditandai sebagai “Siap untuk ditinjau”. Permintaan tarik draf tidak dianalisis.
- Temuan keamanan diposting sebagai komentar ulasan langsung pada permintaan tarik dengan panduan remediasi khusus.
- Analisis menggunakan pengaturan tinjauan kode yang dikonfigurasi (kerentanan keamanan, persyaratan khusus, atau keduanya).

Note

Komentar permintaan tarik hanya tersedia untuk GitHub repositori pribadi.

Saat remediasi Kode diaktifkan untuk repositori, pengguna aplikasi web dapat memulai remediasi untuk tinjauan kode dan temuan pengujian penetrasi pada repositori tersebut, dan AWS Security Agent memberikan setiap perbaikan sebagai permintaan tarik. Untuk informasi selengkapnya, lihat

the section called “Memungkinkan pengguna untuk memulai remediasi uji penetrasi dan temuan tinjauan kode”.

Untuk informasi lebih lanjut tentang bagaimana temuan permintaan tarik muncul GitHub dan bagaimana menanggapinya, lihatthe section called “Tinjau temuan keamanan kode dalam permintaan tarik”.

Tambahkan ember S3

Tambahkan bucket S3 yang berisi kode sumber atau sumber daya kontekstual untuk peninjauan kode.

1. Di bagian bucket S3, pilih Tambahkan sumber daya S3.
2. Masukkan URI S3 untuk bucket atau awalan yang berisi kode sumber Anda.
3. Pilih Tambahkan.

> **Note**
>
> Anda dapat menambahkan hingga 10 sumber daya S3. Bucket S3 dibagi di seluruh kemampuan termasuk tinjauan kode dan pengujian penetrasi.

> **Tip**
>
> Anda dapat menambahkan bucket S3 yang berisi kode sumber, file konfigurasi, templat infrastruktur-sebagai-kode, atau artefak lain yang ingin AWS Security Agent analisis untuk masalah keamanan.

Konfigurasikan pengaturan peninjauan kode

Konfigurasikan jenis masalah keamanan yang dianalisis AWS Security Agent selama tinjauan kode. Pengaturan ini berlaku untuk semua repositori dan sumber dengan peninjauan kode diaktifkan di Ruang Agen ini.

1. Di bagian Pengaturan peninjauan kode, pilih salah satu opsi berikut:
   - Validasi persyaratan keamanan — Validasi apakah kode sesuai dengan persyaratan keamanan yang telah Anda aktifkan dalam paket persyaratan keamanan Anda.

- Temuan kerentanan keamanan - Identifikasi kerentanan keamanan umum dalam kode.
- Persyaratan keamanan dan temuan kerentanan — Analisis kode untuk kepatuhan terhadap persyaratan keamanan yang telah Anda aktifkan dan kerentanan keamanan umum. Ini adalah pengaturan default.

> **Note**
>
> Saat validasi persyaratan keamanan diaktifkan, AWS Security Agent memeriksa kode terhadap persyaratan keamanan yang telah Anda aktifkan dalam paket persyaratan keamanan Anda. Jika Anda memilih validasi persyaratan keamanan tetapi tidak mengaktifkan setidaknya satu persyaratan keamanan, AWS Security Agent tidak akan mengidentifikasi temuan berbasis persyaratan. Untuk informasi selengkapnya tentang persyaratan keamanan, lihatthe section called “Kelola persyaratan keamanan”.

1. Pilih Berikutnya untuk melanjutkan ke konfigurasi opsional.

Langkah 2: Konfigurasi opsional

Pada langkah kedua wizard, konfigurasikan pengaturan CloudWatch pencatatan opsional dan akses layanan untuk lingkungan peninjauan kode Anda.

(Opsional) Konfigurasikan CloudWatch log

Konfigurasikan grup CloudWatch log untuk menangkap dan menganalisis perilaku aplikasi selama peninjauan kode.

1. Perluas bagian CloudWatch log.
2. Di menu tarik-turun Grup Log, pilih satu atau beberapa grup CloudWatch log yang ada dari akun AWS Anda.

> **Note**
>
> Jika Anda tidak memilih grup log, AWS Security Agent secara otomatis membuat grup log default untuk menyimpan log eksekusi peninjauan kode.

(Opsional) Konfigurasikan akses layanan

Konfigurasikan peran layanan IAM yang digunakan AWS Security Agent untuk mengakses sumber daya AWS Anda seperti bucket dan CloudWatch log S3 untuk peninjauan kode.

1. Perluas bagian Akses layanan.
2. Pilih salah satu opsi berikut:
   - Buat peran default — AWS Security Agent secara otomatis membuat peran IAM baru dengan izin yang diperlukan untuk peninjauan kode.
   - Gunakan peran layanan yang ada — Pilih peran IAM yang ada dari menu tarik-turun.
3. Jika menggunakan peran default, masukkan nama peran Layanan. Nama harus unik di semua peran dalam akun, menggunakan karakter dan `+=,.@-_` karakter alfanumerik, dan tidak dapat menyertakan spasi.

> **Note**
>
> Nama peran layanan memiliki panjang maksimum 64 karakter. Peran layanan dibuat secara otomatis jika Anda tidak memilih peran yang ada.

1. Pilih Simpan untuk menyelesaikan pengaturan.

Setelah pengaturan

Setelah menyelesaikan wizard penyiapan peninjauan kode:

- Kartu peninjauan Kode di halaman Ruang Agen Anda menunjukkan status Siap.
- Pengguna dapat meluncurkan aplikasi web dan membuat ulasan kode untuk memindai repositori yang terhubung dan sumber S3.
- Anda dapat mengubah konfigurasi kapan saja dengan memilih Edit konfigurasi pada tab Tinjauan kode.

Edit konfigurasi tinjauan kode

Untuk mengubah konfigurasi peninjauan kode Anda setelah penyiapan awal:

1. Arahkan ke Ruang Agen Anda di konsol AWS Security Agent.
2. Pilih tab Ulasan kode.
3. Pilih Edit konfigurasi.
4. Perbarui integrasi, bucket S3, pengaturan peninjauan kode, CloudWatch log, atau akses layanan sesuai kebutuhan.
5. Pilih Simpan.

Langkah selanjutnya

Setelah menyiapkan konfigurasi peninjauan kode:

- Luncurkan aplikasi web untuk membuat dan menjalankan ulasan kode (lihatthe section called “Buat ulasan kode”)
- Hubungkan GitHub repositori tambahan atau bucket S3 saat basis kode Anda tumbuh
- Konfigurasikan paket persyaratan keamanan untuk validasi khusus organisasi (lihat) the section called “Kelola persyaratan keamanan”
- Tinjau bagaimana temuan permintaan tarik muncul di GitHub (lihatthe section called “Tinjau temuan keamanan kode dalam permintaan tarik”)

Aktifkan pemodelan ancaman

Konfigurasikan Ruang Agen Anda untuk mengaktifkan pemodelan ancaman dengan menghubungkan repositori kode sumber dan mengonfigurasi sumber daya AWS. Pemodelan ancaman menganalisis arsitektur aplikasi Anda dan mengidentifikasi ancaman keamanan dari kode sumber, dokumen desain, atau keduanya.

Menyiapkan konfigurasi pemodelan ancaman adalah Space-wide operasi Agen. Integrasi dan bucket S3 yang Anda sambungkan dibagi di seluruh kemampuan, termasuk pemodelan ancaman, peninjauan kode, dan pengujian penetrasi.

Setelah menyelesaikan penyiapan, pengguna dapat membuat dan menjalankan model ancaman di aplikasi web AWS Security Agent.

> **Note**
>
> Jika Anda sudah memiliki repositori atau bucket S3 yang terhubung ke Ruang Agen Anda (misalnya, melalui peninjauan kode atau pengaturan pengujian penetrasi), pemodelan

ancaman mungkin sudah diaktifkan. Anda dapat langsung menuju ke aplikasi web untuk membuat model ancaman. Lihat the section called “Buat model ancaman”.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Ruang Agen yang dibuat di AWS Management Console (lihatthe section called “Buat Ruang Agen”)
- Izin untuk mengonfigurasi integrasi untuk Ruang Agen Anda
- (Opsional) Organisasi GitHub, GitLab, atau Bitbucket dengan aplikasi AWS Security Agent diinstal (lihat Connect AWS Security Agent ke repositori, Connect AWS Security Agent ke GitHub repositori, atau Connect AWS Security Agent ke GitLab repositori Bitbucket)

Akses wizard penyiapan pemodelan ancaman

Arahkan ke konfigurasi pemodelan ancaman untuk Ruang Agen Anda.

1. Di konsol AWS Security Agent, pilih Ruang Agen Anda.
2. Pilih Konfigurasikan model ancaman dari kartu model Ancaman, atau dari tab Model ancaman.

Anda diarahkan ke wizard Configure threat model, yang memiliki dua langkah opsional.

Langkah 1: Connect repositori kode sumber (opsional)

Hubungkan repositori GitHub GitLab,, atau Bitbucket yang ingin Anda aktifkan pemodelan ancaman. Model ancaman itu sendiri dibuat dan dilihat dalam aplikasi web.

> **Important**
>
> Integrasi yang dikonfigurasi di sini dibagikan di seluruh Ruang Agen Anda. Perubahan berlaku untuk pemodelan ancaman, tinjauan kode, dan kemampuan pengujian penetrasi.

Connect repositori

1. Di bagian Integrasi terhubung, pilih Tambah.
2. Pilih registrasi yang berisi repositori yang ingin Anda gunakan.

3. Pilih kotak centang untuk setiap repositori yang ingin Anda sambungkan.
4. Pilih Simpan untuk menerapkan pilihan Anda.

> **Note**
>
> Jika Anda belum mendaftarkan integrasi, pilih Pengaturan untuk menavigasi ke halaman Integrasi tempat Anda dapat mengotorisasi aplikasi AWS Security Agent. Untuk informasi selengkapnya, lihat Connect AWS Security Agent ke GitHub repositori, Connect AWS Security Agent ke GitLab repositori, atau Connect AWS Security Agent ke repositori Bitbucket.

1. Pilih Berikutnya untuk melanjutkan ke konfigurasi opsional.

Langkah 2: Konfigurasi opsional

Konfigurasikan pengaturan bucket, CloudWatch logging, dan akses layanan S3 untuk lingkungan pemodelan ancaman Anda. Pengaturan ini dibagikan dengan kemampuan lain di Ruang Agen Anda.

Ember S3 (opsional)

Tambahkan bucket S3 yang berisi kode sumber yang Anda ingin agen gunakan sebagai konteks selama pemodelan ancaman.

1. Di bagian bucket S3, pilih Tambahkan sumber daya S3.
2. Masukkan URI S3 untuk bucket atau awalan.
3. Pilih Tambahkan.

> **Note**
>
> Anda dapat menambahkan hingga 10 sumber daya S3.

CloudWatch log (opsional)

Konfigurasikan grup CloudWatch log untuk menangkap dan menganalisis perilaku aplikasi selama model ancaman berjalan.

1. Di bagian CloudWatch log, pilih satu atau beberapa grup CloudWatch log yang ada dari akun AWS Anda.

Akses layanan

Konfigurasikan peran layanan IAM yang digunakan AWS Security Agent untuk mengakses sumber daya AWS Anda seperti bucket dan CloudWatch log S3 untuk pemodelan ancaman. Peran layanan diperlukan untuk mengaktifkan pemodelan ancaman.

1. Di bagian Akses layanan, pilih peran layanan IAM yang ada dari menu tarik-turun, atau biarkan kosong agar peran layanan dibuat secara otomatis.

> **Note**
>
> Peran layanan akan dibuat secara otomatis jika Anda tidak memilih peran yang ada.

1. Pilih Simpan untuk menyelesaikan konfigurasi.

Setelah pengaturan

Setelah menyelesaikan konfigurasi pemodelan ancaman:

- Kartu model Ancaman di halaman Ruang Agen Anda menunjukkan status Siap.
- Pengguna dapat meluncurkan aplikasi web dan membuat model ancaman dari kode sumber yang terhubung, dokumen lingkup yang diunggah, halaman Confluence, atau kombinasi input ini.

Langkah selanjutnya

Setelah mengaktifkan pemodelan ancaman:

- Luncurkan aplikasi web untuk membuat dan menjalankan model ancaman (lihat Membuat model ancaman)
- Hubungkan repositori tambahan atau bucket S3 saat basis kode Anda tumbuh
- Connect Confluence untuk mengaktifkan pemilihan halaman sebagai dokumen cakupan (lihat Connect AWS Security Agent to Confluence)

Memungkinkan pengguna untuk memulai remediasi uji penetrasi dan temuan tinjauan kode

Di AWS Management Console, Anda dapat mengaktifkan remediasi otomatis sehingga pengguna aplikasi web AWS Security Agent dapat meminta perbaikan untuk temuan tertentu. AWS Security Agent memberikan setiap perbaikan sebagai permintaan GitHub tarik pada repositori yang terpengaruh.

Anda mengaktifkan remediasi per repositori dari dalam Ruang Agen. Tab Penetration test dan Code review membuka wizard konfigurasi repositori yang sama, sehingga Anda dapat menggunakan salah satu tab untuk mengelola pengaturan Automatic remediation enabled. Remediasi berlaku untuk temuan dari kedua kemampuan, jadi Anda hanya perlu mengaktifkannya sekali per repositori.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

1. Pengujian penetrasi yang diaktifkan (lihatthe section called “Aktifkan uji penetrasi”) atau tinjauan kode (lihatthe section called “Aktifkan peninjauan kode”)
2. Menginstal dan mengesahkan GitHub Aplikasi AWS Security Agent untuk GitHub organisasi Anda (lihatthe section called “Connect AWS Security Agent ke GitHub repositori”)

Buka wizard konfigurasi repositori

Pilih titik masuk yang cocok dengan cara repositori Anda diatur.

Dari tab Tes Penetrasi

Gunakan jalur ini untuk menambahkan GitHub repositori untuk pengujian penetrasi dan mengonfigurasi remediasi di pass yang sama.

1. Arahkan ke halaman ikhtisar Ruang Agen.
2. Pilih tab Tes Penetrasi.
3. Pilih GitHub registrasi yang memiliki repositori Anda:
   a. Jika Anda belum mengaitkan GitHub pendaftaran apa pun dengan Ruang Agen, pilih Tambahkan di kotak informasi Connect GitHub to AWS Security Agent untuk memilih pendaftaran.

b. Jika satu atau beberapa GitHub pendaftaran sudah terkait, pilih Tambah di bagian Integrasi terhubung untuk memilih yang lain.

4. Pilih Berikutnya untuk memilih GitHub repositori.
5. Pilih Berikutnya untuk mengkonfigurasi kemampuan repositori.

Dari tab Ulasan kode

Gunakan jalur ini ketika repositori Anda sudah terhubung untuk peninjauan kode.

1. Arahkan ke halaman ikhtisar Ruang Agen.
2. Pilih tab Ulasan kode.
3. Di tabel GitHub repositori, pilih Edit untuk membuka wizard konfigurasi repositori.

Aktifkan remediasi otomatis dan simpan

Setelah Anda mencapai kemampuan repositori, lanjutkan melalui salah satu jalur di atas:

1. Di kolom Automatic remediation enabled, tandai setiap repositori yang ingin diperbaiki sebagai Diaktifkan.
2. Pilih Connect untuk menyimpan konfigurasi.

Pengguna aplikasi web sekarang dapat memulai remediasi untuk temuan di repositori ini, dan AWS Security Agent akan membuka permintaan tarik dengan perbaikan yang diusulkan.

Menyediakan sumber daya agen dari bucket S3

Anda dapat memberikan AWS Security Agent akses ke sumber daya dalam bucket S3, seperti dokumen API, dokumen model ancaman, dan kode sumber yang mendukung pengujian penetrasi.

Anda memberikan akses baca AWS Security Agent ke bucket S3 di AWS Management Console. Di aplikasi web Agen Keamanan, pengguna memilih sumber daya individual dari S3 sebagai relevan.

1. Arahkan ke halaman ikhtisar Ruang Agen
2. Pilih Tindakan dan kemudian Edit konfigurasi uji penetrasi
3. Di bawah bagian bucket S3, tentukan URI S3 yang berisi Bucket S3. Anda dapat menambahkan beberapa ember S3.

Aktifkan domain aplikasi untuk pengujian penetrasi

Sebelum Anda dapat menjalankan tes penetrasi pada aplikasi, Anda perlu menambahkan domain target dan memverifikasi kepemilikan. AWS Security Agent hanya akan melakukan pengujian penetrasi terhadap domain terverifikasi.

> **Note**
>
> Anda tidak perlu memvalidasi domain tambahan yang mungkin digunakan aplikasi Anda. Anda hanya perlu memvalidasi domain yang akan Anda jalankan secara aktif dalam tes penetrasi.

Langkah 1: Tambahkan domain target

Untuk menambahkan domain target, navigasikan ke tab Uji Penetrasi di halaman ringkasan Ruang Agen. Bergantung pada apakah Anda telah mengonfigurasi pengujian penetrasi, gunakan salah satu metode berikut:

- First-time setup: Pilih Siapkan tes penetrasi untuk membuka wizard uji penetrasi. Pada langkah pertama, masukkan domain dan pilih metode verifikasi.
- Menambahkan domain ke konfigurasi yang ada: Di tabel Domain Target, pilih Tambahkan domain. Masukkan domain dan pilih metode verifikasi di modal.

Anda dapat menambahkan domain dasar atau sub-domain, seperti `example.com` atau`billing.example.com`. AWS menyarankan untuk menggunakan sub-domain tempat Anda memiliki izin untuk membuat `TXT` catatan.

> **Note**
>
> Ketika Anda memverifikasi domain menggunakan DNS TXT atau verifikasi rute HTTP, sub-domain domain tersebut secara otomatis tercakup. Misalnya, verifikasi `example.com` memungkinkan Anda untuk menguji `api.example.com` atau `billing.example.com` tanpa verifikasi tambahan. Untuk verifikasi VPC Pribadi, nama domain titik akhir target harus sesuai dengan nama domain lengkap yang dikonfigurasi untuk verifikasi.

Pilih salah satu metode verifikasi berikut:

- Catatan DNS TXT: Buktikan kepemilikan domain dengan membuat catatan DNS TXT dengan penyedia DNS Anda.
- Rute HTTP: Buktikan kepemilikan domain dengan membuat rute di server web Anda yang berisi token unik yang disediakan oleh AWS Security Agent.
- VPC pribadi: Hanya dapat digunakan untuk pengujian penetrasi VPC pribadi. Memverifikasi bahwa domain menyelesaikan ke IP dalam rentang CIDR pribadi. Nama domain yang dikonfigurasi harus sesuai dengan nama domain lengkap dari setiap titik akhir target terkait

Langkah 2: Verifikasi kepemilikan domain

Setelah menambahkan domain, verifikasi kepemilikan menggunakan metode yang Anda pilih. Anda dapat memicu verifikasi kapan saja dari tabel Domain Target dengan memilih domain dan memilih Verifikasi.

Verifikasi menggunakan catatan DNS TXT

AWS Security Agent menghasilkan nilai catatan DNS TXT. Anda harus menambahkan catatan ini dengan penyedia DNS Anda untuk membuktikan kepemilikan.

Jika domain terdaftar di Route 53 (akun AWS yang sama):

AWS Security Agent dapat membuat catatan DNS secara otomatis. Pilih domain dari tabel Domain Target dan pilih One-click verifikasi. AWS Security Agent membuat catatan validasi DNS dan menyelesaikan verifikasi secara otomatis.

Jika domain terdaftar dengan penyedia DNS lain:

1. Salin nilai catatan TXT yang disediakan oleh AWS Security Agent.
2. Tambahkan catatan TXT dengan registrar DNS Anda.
3. Kembali ke tabel Domain Target, pilih domain, dan pilih Verifikasi.

Verifikasi menggunakan validasi rute HTTP

Metode ini membuktikan kepemilikan domain dengan menempatkan token unik di server web Anda. Hanya pemilik domain atau administrator web resmi yang dapat membuat rute di server web, yang membuktikan kepemilikan.

1. Buat file di jalur berikut di server web Anda:

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

2. Tempatkan token yang disediakan oleh AWS Security Agent dalam file menggunakan format ini:

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

3. Kembali ke tabel Domain Target, pilih domain, dan pilih Verifikasi. AWS Security Agent mengirimkan permintaan HTTPS GET ke URL verifikasi dan memvalidasi token.
4. Jika domain dapat diakses di internet publik, pastikan domain Anda memiliki sertifikat SSL yang valid sebelum menjalankan verifikasi.

> **Note**
>
> Jika domain Anda terdaftar di beberapa ruang agen dan Anda menggunakan validasi rute HTTP, Anda dapat menempatkan token untuk kedua ruang agen dalam `tokens` array yang sama.

Lewati verifikasi kepemilikan domain

Pelanggan yang memiliki otorisasi untuk melakukan pengujian penetrasi pada titik akhir tetapi tidak dapat menyelesaikan verifikasi kepemilikan dapat meminta verifikasi manual dari kami. Silakan buka kasus dukungan pelanggan untuk mengajukan permintaan ini. Permintaan Anda harus menyertakan kasus penggunaan bisnis Anda dan pembenaran Anda untuk verifikasi kepemilikan manual (yaitu, mengapa Anda berwenang untuk melakukan pengujian penetrasi pada titik akhir).

Domain target terkelola yang digunakan untuk pengujian penetrasi

Di AWS Management Console, Anda dapat menambahkan dan mengelola domain target yang digunakan oleh ruang agen untuk pengujian penetrasi. Domain target ini perlu diverifikasi sebelum dapat digunakan dalam tes penetrasi. Untuk informasi selengkapnya tentang memverifikasi domain target, lihat the section called “Aktifkan uji penetrasi”

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

1. Tes penetrasi yang diaktifkan (lihatthe section called “Aktifkan uji penetrasi”)

Kelola sumber daya domain target

- Arahkan ke halaman ikhtisar Domain Target.
- Anda harus melihat semua sumber daya domain target yang terkait dengan akun Anda
  - Jika Anda tidak melihat domain target yang terkait dengan akun Anda, ikuti langkah-langkah the section called “Aktifkan uji penetrasi” untuk membuat dan memverifikasi domain target
- Domain target dapat digunakan kembali antara ruang agen dan status verifikasi berbagi
  - Untuk menambahkan domain target yang ada ke ruang agen, navigasikan ke tab Uji Penetrasi ruang agen. Pilih Tambahkan domain dan pilih domain yang diinginkan di bawah Pilih dari domain terdaftar sebelumnya yang tersedia di bidang nama domain
  - Domain target harus dikaitkan dengan ruang agen sebelum dapat digunakan dalam uji penetrasi
- Menghapus domain dari ruang agen tidak menghapus domain. Domain target terkait dapat dihapus secara permanen dari halaman ikhtisar Domain Target

Verifikasi domain target

Agar domain target dapat digunakan dalam uji penetrasi, domain tersebut harus diverifikasi terlebih dahulu menggunakan salah satu metode di bawah ini:

- Route 53 domain (akun AWS yang sama): Pilih One-click verifikasi. AWS Security Agent secara otomatis membuat catatan DNS dan menyelesaikan verifikasi.
- DNS TXT (penyedia DNS lainnya): Salin token verifikasi, tambahkan catatan TXT dengan registrar DNS Anda, lalu pilih domain dan pilih Verifikasi.
- Rute HTTP: Tempatkan token verifikasi di jalur rute yang diperlukan di server web Anda, lalu pilih domain dan pilih Verifikasi. Lihat perinciannya di the section called “Aktifkan domain aplikasi untuk pengujian penetrasi”.
- VPC Pribadi: Memverifikasi IP domain target berada dalam rentang CIDR pribadi (lihat the section called “Connect agen ke sumber daya VPC pribadi” untuk daftar rentang CIDR pribadi). Nama domain titik akhir target pengujian penetrasi harus sesuai dengan nama domain lengkap yang dikonfigurasi untuk verifikasi. Hanya dapat digunakan untuk pengujian penetrasi VPC pribadi

> **Note**
>
> Untuk verifikasi DNS TXT, rute HTTP, dan Route 53, sub-domain dari domain terverifikasi secara otomatis tercakup dan tidak memerlukan verifikasi terpisah. Untuk verifikasi VPC Pribadi, setiap domain harus diverifikasi satu per satu.

Kelola persyaratan keamanan

Tentukan persyaratan keamanan yang dievaluasi AWS Security Agent selama tinjauan desain dan kode, menggunakan AWS-managed paket, paket khusus, atau persyaratan yang diimpor dari dokumentasi Anda sendiri.

Kelola persyaratan keamanan

Atur persyaratan keamanan yang digunakan Agen AWS Keamanan untuk menganalisis aplikasi Anda ke dalam paket persyaratan keamanan. Paket adalah kumpulan persyaratan keamanan. Anda dapat mengaktifkan paket AWS-managed, membuat paket kustom Anda sendiri, dan menghasilkan persyaratan untuk paket kustom dengan mengunggah dokumentasi keamanan Anda.

Gambaran umum

Persyaratan keamanan menentukan standar atau kebijakan keamanan yang dievaluasi oleh Agen AWS Keamanan terhadap aplikasi Anda selama tinjauan desain dan tinjauan kode. Ketika desain atau kode tidak memenuhi persyaratan, Agen AWS Keamanan melaporkan temuan. Setiap persyaratan keamanan termasuk dalam paket persyaratan keamanan. Persyaratan keamanan dibagikan di semua Ruang Agen dan dievaluasi selama tinjauan desain dan kode.

AWS Agen Keamanan menyediakan dua jenis paket, ditampilkan pada tab terpisah:

- Paket persyaratan keamanan terkelola AWS— paket persyaratan keamanan yang disediakan berdasarkan standar industri dan praktik terbaik. Anda dapat mengaktifkan paket ini secara instan untuk mengevaluasi kebijakan keamanan umum tanpa konfigurasi khusus. Persyaratan dalam paket terkelola adalah hanya-baca. Anda dapat menggunakan persyaratan AWS-managed sebagai template untuk persyaratan kustom. Untuk daftar paket terkelola yang tersedia, lihat paket persyaratan keamanan AWS terkelola.
- Paket persyaratan keamanan khusus — Paket yang Anda buat untuk menegakkan kebijakan dan standar spesifik organisasi Anda. Anda membuat persyaratan dalam paket khusus dari awal,

menyesuaikan persyaratan yang AWS dikelola sebagai titik awal, atau membuatnya dengan mengunggah dokumentasi keamanan Anda.

> **Note**
>
> Paket kerangka kepatuhan dirancang untuk membantu mengidentifikasi persyaratan keamanan yang mungkin relevan untuk kepatuhan dengan kerangka kerja tertentu. Mereka tidak menilai kepatuhan Anda atau menjamin bahwa Anda akan lulus audit.

Anda mengaktifkan dan menonaktifkan paket sebagai satu unit. Saat paket diaktifkan, Agen AWS Keamanan mengevaluasi aplikasi Anda terhadap persyaratan dalam paket tersebut selama peninjauan desain dan kode.

> **Note**
>
> Mengaktifkan atau menonaktifkan paket berlaku untuk ulasan desain baru dan ulasan kode. Ulasan yang ada tidak terpengaruh.

Mengaktifkan atau menonaktifkan paket terkelola

AWS Aktifkan paket yang dikelola untuk mengevaluasi aplikasi Anda terhadap serangkaian persyaratan keamanan yang AWS dikelola. Nonaktifkan saat Anda tidak lagi membutuhkannya.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan terkelola.
4. Pilih paket yang ingin Anda aktifkan atau nonaktifkan.
5. Lakukan salah satu tindakan berikut:
   a. Untuk mengaktifkan paket, pilih Aktifkan paket.
   b. Untuk menonaktifkan paket, pilih Nonaktifkan paket.

> **Tip**
>
> Pilih paket untuk melihat persyaratan keamanan yang dikandungnya. Pilih persyaratan untuk melihat definisi lengkapnya, termasuk penerapannya, kriteria kepatuhan, dan panduan remediasi.

Buat paket khusus

Buat paket khusus untuk mengelompokkan persyaratan keamanan yang khusus untuk organisasi Anda. Setelah Anda membuat paket, tambahkan persyaratan secara manual, sesuaikan persyaratan yang AWS dikelola, atau unggah dokumen untuk menghasilkan persyaratan.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus.
4. Pilih Buat paket persyaratan keamanan.
5. Masukkan nama paket persyaratan keamanan dan deskripsi opsional.
6. Pilih Buat paket persyaratan keamanan.

> **Note**
>
> Setelah membuat paket, Anda dapat menambahkan atau mengunggah dokumen sumber untuk menghasilkan persyaratan keamanan untuk paket Anda.

Tambahkan persyaratan keamanan secara manual

Tambahkan persyaratan keamanan ke paket khusus untuk menentukan standar keamanan yang diberlakukan Agen AWS Keamanan.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus, lalu pilih paket yang ingin Anda tambahkan persyaratan.
4. Pilih Tambahkan persyaratan secara manual.

5. Konfigurasi bidang berikut:
   a. Nama persyaratan keamanan — Masukkan nama deskriptif yang mengidentifikasi kontrol keamanan, misalnya `enforce-encryption-at-rest` (maksimum 80 karakter).
   b. Deskripsi — Berikan deskripsi singkat tentang apa yang diperiksa persyaratan keamanan ini (maksimum 500 karakter).
   c. Penerapan — Jelaskan skenario, jenis sistem, atau kondisi di mana persyaratan keamanan ini harus dievaluasi. Sertakan skenario di mana persyaratan harus ditandai NOT_APPLICABLE (maksimum 10.000 karakter).
   d. Kriteria kepatuhan — Tentukan apa yang dimaksud dengan kepatuhan versus ketidakpatuhan terhadap persyaratan keamanan ini. Berikan indikator, contoh, dan detail teknis yang AWS dicari Agen Keamanan saat mengevaluasi kepatuhan (maksimum 10.000 karakter).
   e. Panduan Remediasi (Opsional) — Berikan panduan tentang cara memperbaiki pelanggaran, termasuk tautan ke dokumentasi atau standar internal organisasi Anda (maksimum 10.000 karakter).
6. Lakukan salah satu tindakan berikut:
   a. Untuk membuat persyaratan tanpa mengaktifkannya, pilih Buat persyaratan keamanan.
   b. Untuk membuat persyaratan dan mengaktifkannya, pilih Buat dan aktifkan persyaratan keamanan.

> **Note**
>
> Persyaratan dievaluasi selama tinjauan keamanan hanya ketika paket yang berisi itu diaktifkan. Untuk mengontrol apakah paket dievaluasi, aktifkan atau nonaktifkan paket.

Kustomisasi AWS-Persyaratan keamanan terkelola

Buat persyaratan keamanan khusus yang menggunakan persyaratan AWS-managed sebagai titik awalnya. AWS Agen Keamanan mengisi detail persyaratan dari persyaratan terkelola, dan Anda mengeditnya agar sesuai dengan organisasi Anda.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus, lalu pilih paket khusus untuk menambahkan persyaratan.

4. Pilih Buat persyaratan keamanan khusus.
5. Untuk mengisi formulir sebelumnya, di bagian Sesuaikan persyaratan keamanan yang AWS dikelola, cari dan pilih persyaratan yang AWS dikelola untuk digunakan sebagai templat.
6. Edit salah satu bidang agar sesuai dengan kebutuhan organisasi Anda.
7. Lakukan salah satu tindakan berikut:
   a. Untuk membuat persyaratan tanpa mengaktifkannya, pilih Buat persyaratan keamanan.
   b. Untuk membuat dan segera mengaktifkan persyaratan, pilih Buat dan aktifkan persyaratan keamanan.

> **Note**
>
> Menyesuaikan persyaratan AWS-managed menciptakan persyaratan keamanan kustom independen. Perubahan selanjutnya pada persyaratan AWS-managed tidak memengaruhi versi kustom Anda.

Hasilkan persyaratan dengan mengunggah dokumen

Hasilkan persyaratan keamanan untuk paket khusus dari dokumentasi keamanan Anda yang ada alih-alih menulis setiap persyaratan dengan tangan. AWS Agen Keamanan membaca dokumen yang Anda unggah, mengidentifikasi konten yang relevan dengan keamanan, dan menghasilkan persyaratan terstruktur yang dapat Anda tinjau dan edit. Untuk prosedur selengkapnya, lihat Menghasilkan persyaratan keamanan dari dokumen. Untuk panduan tentang apa yang harus disertakan dalam dokumen yang Anda unggah, lihat Mempersiapkan dokumen untuk pembuatan persyaratan.

> **Important**
>
> Mengunggah dokumen sumber baru akan meregenerasi semua persyaratan untuk paket. Persyaratan apa pun yang ada, termasuk yang Anda tambahkan secara manual, diganti.

Mengedit persyaratan keamanan khusus

Ubah persyaratan keamanan khusus untuk memperbarui definisi, kriteria, atau panduan remediasinya. Anda tidak dapat mengedit persyaratan dalam paket AWS-managed.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus, lalu pilih paket yang berisi persyaratan.
4. Pilih persyaratan yang ingin Anda edit, lalu pilih Edit persyaratan keamanan khusus.
5. Perbarui bidang persyaratan sesuai kebutuhan. Nama persyaratan keamanan tidak dapat diubah setelah pembuatan.
6. Pilih Perbarui persyaratan keamanan.

> **Note**
>
> Perubahan pada persyaratan keamanan berlaku untuk ulasan desain baru dan ulasan kode. Ulasan yang ada tidak terpengaruh.

Mengaktifkan atau menonaktifkan paket

Mengaktifkan atau menonaktifkan paket persyaratan keamanan untuk mengontrol apakah Agen AWS Keamanan mengevaluasi persyaratan yang dikandungnya. Anda mengaktifkan dan menonaktifkan paket sebagai satu unit.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab untuk jenis paket, lalu pilih paket.
4. Lakukan salah satu tindakan berikut:
   a. Untuk mengaktifkan paket, pilih Aktifkan paket.
   b. Untuk menonaktifkan paket, pilih Nonaktifkan paket.

> **Note**
>
> Saat paket diaktifkan, persyaratan dalam paket dievaluasi selama tinjauan keamanan. Ketika paket dinonaktifkan, persyaratannya tidak dievaluasi.

Hapus persyaratan keamanan khusus

Hapus persyaratan keamanan khusus saat Anda tidak lagi ingin Agen AWS Keamanan mengevaluasinya.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus, lalu pilih paket yang berisi persyaratan.
4. Pilih satu atau beberapa persyaratan keamanan, lalu pilih Hapus persyaratan keamanan.
5. Konfirmasi penghapusan.

> **Note**
>
> Ketika Anda menghapus persyaratan keamanan, mereka tidak lagi dievaluasi dalam tinjauan masa depan. Persyaratan tetap terlihat di ulasan sebelumnya sebagai hasil hanya-baca.

Praktik terbaik untuk mendefinisikan persyaratan keamanan

Saat Anda membuat persyaratan keamanan, ikuti panduan ini untuk membantu Agen AWS Keamanan mengevaluasi aplikasi Anda secara akurat dan memberikan temuan yang dapat ditindaklanjuti.

Nama persyaratan keamanan — Gunakan nama yang jelas dan spesifik yang mengidentifikasi kontrol keamanan. Hindari istilah generik yang tidak menyampaikan tujuan persyaratan.

Deskripsi — Jelaskan apa yang diperiksa kontrol dan risiko yang dimitigasi. Ini membantu pengguna memahami mengapa persyaratan itu penting.

Penerapan — Jelaskan beban kerja, jenis sistem, atau kondisi di mana persyaratan harus dievaluasi. Nyatakan kapan persyaratan harus ditandai NOT_APPLICABLE, dengan skenario tertentu, untuk menghindari positif palsu. Frasa seperti “Kontrol ini berlaku untuk SEMUA beban kerja yang...” dan “Tandai sebagai NOT_APPLICABLE jika...” menetapkan batas cakupan yang jelas.

Kriteria kepatuhan - Susun ini dalam dua bagian: apa yang menunjukkan kepatuhan dan apa yang menunjukkan ketidakpatuhan. Jadilah spesifik dengan indikator teknis dan sertakan kasus tepi. Mulailah dengan “Desain sesuai jika menunjukkan...” diikuti dengan detail teknis, lalu “Desain tidak sesuai jika...” dengan pola yang menunjukkan pelanggaran.

Panduan remediasi - Berikan panduan dengan detail teknis dan contoh konfigurasi. Sertakan tautan ke dokumentasi internal organisasi Anda sehingga pengembang memiliki sumber daya yang mereka butuhkan untuk memperbaiki pelanggaran.

Langkah selanjutnya

Setelah Anda mengonfigurasi paket persyaratan keamanan Anda:

- Buat tinjauan desain atau tinjauan kode untuk mengevaluasi aplikasi Anda terhadap paket yang Anda aktifkan.
- Aktifkan pengujian penetrasi untuk melengkapi desain dan ulasan kode Anda.
- Berikan pengguna akses ke aplikasi web Agen AWS Keamanan.

Paket persyaratan keamanan terkelola AWS

Paket persyaratan keamanan AWS terkelola adalah kumpulan persyaratan keamanan yang AWS menciptakan dan memelihara, berdasarkan standar industri dan praktik terbaik. Anda mengaktifkan paket terkelola untuk mengevaluasi aplikasi Anda terhadap persyaratannya selama tinjauan desain dan ulasan kode, tanpa menulis persyaratan sendiri.

Persyaratan keamanan dalam paket terkelola adalah hanya-baca. Anda tidak dapat mengubahnya. Anda dapat menggunakan persyaratan AWS-managed sebagai template untuk membuat persyaratan kustom, lalu mengedit salinan agar sesuai dengan organisasi Anda. Untuk informasi selengkapnya, lihat Mengelola persyaratan keamanan.

AWS memperbarui paket terkelola dari waktu ke waktu. Saat AWS menambahkan atau merevisi persyaratan dalam paket terkelola, perubahan tersebut berlaku untuk setiap akun yang mengaktifkan paket tersebut.

> **Note**
>
> Paket kerangka kepatuhan dirancang untuk membantu mengidentifikasi persyaratan keamanan yang mungkin relevan untuk kepatuhan dengan kerangka kerja tertentu. Mereka tidak menilai kepatuhan Anda atau menjamin bahwa Anda akan lulus audit.

Paket terkelola AWS: Paket Dasar ASA

Seperangkat persyaratan keamanan dasar yang berlaku untuk sebagian besar beban kerja. Paket ini mencakup masalah keamanan aplikasi umum seperti otentikasi dan otorisasi, perlindungan data, logging, dan validasi input. Aktifkan paket ini untuk menetapkan garis dasar keamanan untuk desain dan ulasan kode Anda.

Paket terkelola AWS: AWS Well-Architected Pack

Persyaratan keamanan berasal dari pilar keamanan AWS Well-Architected Kerangka Kerja. Paket ini mengevaluasi aplikasi Anda terhadap AWS panduan untuk melindungi data, mengelola identitas dan izin, serta mendeteksi dan menanggapi peristiwa keamanan. Untuk informasi selengkapnya tentang kerangka kerja, lihat Pilar Keamanan - AWS Well-Architected Framework.

Paket terkelola AWS: Paket CSF NIST

Persyaratan keamanan berasal dari NIST Cybersecurity Framework (CSF). Paket ini mengevaluasi aplikasi Anda terhadap area kontrol yang diambil dari kerangka kerja, seperti manajemen identitas dan akses, keamanan data, dan teknologi pelindung. Aktifkan paket ini untuk menyelaraskan desain dan ulasan kode Anda dengan NIST CSF.

Paket terkelola AWS: Paket PCI DSS

Persyaratan keamanan berasal dari Payment Card Industry Data Security Standard (PCI DSS). Paket ini mengevaluasi aplikasi Anda terhadap area kontrol yang relevan dengan penanganan data pemegang kartu, seperti kontrol akses, enkripsi data saat transit dan saat istirahat, dan pencatatan. Aktifkan paket ini untuk menyelaraskan desain dan ulasan kode Anda dengan PCI DSS.

Hasilkan persyaratan keamanan dari dokumen

Hasilkan persyaratan keamanan untuk paket kustom dengan mengunggah dokumentasi keamanan yang ada. AWS Agen Keamanan membaca dokumen yang Anda unggah, mengidentifikasi konten yang relevan dengan keamanan, dan menghasilkan persyaratan keamanan terstruktur dengan penerapan, kriteria kepatuhan, dan panduan remediasi. Anda dapat meninjau dan mengedit persyaratan yang dihasilkan sebelum digunakan dalam ulasan.

Gunakan ini alih-alih menulis setiap persyaratan dengan tangan ketika Anda sudah memiliki kebijakan keamanan, standar, atau pedoman yang didokumentasikan. Untuk panduan tentang apa

yang harus disertakan dalam dokumen yang Anda unggah, lihat Mempersiapkan dokumen untuk pembuatan persyaratan.

Format file yang didukung

Anda dapat mengunggah format file berikut:

- DOC
- DOCX
- MD
- PDF
- TXT

Hasilkan persyaratan

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus.
4. Buat paket, atau pilih paket kustom yang ada untuk menghasilkan persyaratan.
5. Pilih Pilih file, atau seret dan jatuhkan dokumen Anda ke area unggahan.
6. Pilih Hasilkan persyaratan.
7. Tunggu Agen AWS Keamanan untuk memproses dokumen. Generasi berjalan di latar belakang dan dapat memakan waktu beberapa menit untuk file besar. Anda tidak dapat memulai unggahan lain untuk paket saat pembuatan sedang berlangsung.
8. Tinjau persyaratan keamanan yang dihasilkan. Edit, hapus, atau tambahkan persyaratan sesuai kebutuhan. Aktifkan paket saat Anda ingin Agen AWS Keamanan mengevaluasi persyaratannya selama tinjauan keamanan.

> **Note**
>
> AWS Agen Keamanan menghasilkan persyaratan di tingkat beban kerja dan berfokus pada konten yang relevan dengan keamanan dalam dokumen Anda. Jumlah persyaratan yang dihasilkannya tergantung pada konten yang Anda berikan. Tinjau persyaratan yang dihasilkan untuk mengonfirmasi bahwa persyaratan tersebut mencerminkan maksud Anda.

Ganti persyaratan dengan unggahan baru

Mengunggah dokumen sumber baru akan meregenerasi persyaratan untuk paket.

> **Important**
>
> Saat Anda mengunggah dokumen sumber baru ke paket, Agen AWS Keamanan meregenerasi semua persyaratan untuk paket tersebut. Persyaratan apa pun yang ada, termasuk yang Anda tambahkan secara manual, diganti. Untuk menjaga kebutuhan Anda saat ini, jangan mengunggah dokumen baru.

1. Di AWS konsol, navigasikan ke Agen AWS Keamanan.
2. Di panel navigasi, pilih Persyaratan keamanan.
3. Pilih tab Paket persyaratan keamanan khusus, lalu pilih paket.
4. Mulai unggahan baru, dan akui bahwa mengunggah dokumen sumber baru menggantikan persyaratan yang ada.
5. Pilih file Anda, lalu pilih Hasilkan persyaratan.

Langkah selanjutnya

- Tinjau dan aktifkan persyaratan yang dihasilkan. Lihat Mengelola persyaratan keamanan.
- Buat tinjauan desain atau tinjauan kode untuk mengevaluasi aplikasi Anda terhadap paket.

Siapkan dokumen untuk pembuatan kebutuhan

Saat Anda membuat persyaratan keamanan dari dokumen, Agen AWS Keamanan membaca dokumen Anda dan menghasilkan persyaratan terstruktur dari konten yang relevan dengan keamanan yang ditemukannya. Anda tidak perlu memformat dokumen Anda dengan cara tertentu atau pra-menulis penerapan, kriteria kepatuhan, dan panduan remediasi. AWS Agen Keamanan mengambil mereka dari konten yang Anda berikan. Unggah dokumentasi keamanan yang sudah Anda miliki, ditulis dengan kata-kata Anda sendiri.

Halaman ini menjelaskan apa yang harus disertakan sehingga Agen AWS Keamanan menghasilkan persyaratan yang akurat dan berguna. Untuk prosedur upload dan batas file, lihat Menghasilkan persyaratan keamanan dari dokumen.

Apa yang harus disertakan

AWS Agen Keamanan mencari konten yang menggambarkan harapan keamanan Anda. Dokumen yang mencakup topik-topik berikut menghasilkan persyaratan terkuat:

- Kontrol akses dan otorisasi
- Autentikasi
- Perlindungan data dan enkripsi
- Keamanan jaringan
- Pencatatan dan audit
- Respons insiden
- Kerentanan dan manajemen tambalan
- Kewajiban kepatuhan yang berlaku untuk beban kerja Anda

Dokumen sumber yang baik termasuk kebijakan keamanan, standar teknik, katalog kontrol, pedoman arsitektur, dan standar pengkodean yang aman.

Menulis di tingkat beban kerja

AWS Agen Keamanan menghasilkan persyaratan yang berlaku untuk beban kerja atau aplikasi individu, ruang lingkup yang sama yang dievaluasi selama tinjauan desain dan kode. Konten yang menjelaskan cara membangun dan mengoperasikan beban kerja yang aman menghasilkan persyaratan. Organization-wide Topik tata kelola, seperti strategi multi-akun, manajemen pengguna root, dan pengaturan pencatatan tingkat akun, berada di luar cakupan ini dan tidak menghasilkan persyaratan beban kerja.

Jelaskan hasilnya, bukan implementasi tunggal

Nyatakan hasil keamanan yang Anda harapkan daripada satu produk atau konfigurasi yang ditentukan. AWS Agen Keamanan menghasilkan persyaratan yang berfokus pada kemampuan keamanan yang dibutuhkan dan memungkinkan lebih dari satu implementasi yang valid. Misalnya, “data saat istirahat dienkripsi” menghasilkan persyaratan yang lebih jelas dan lebih luas daripada pernyataan yang menyebutkan satu layanan atau pengaturan tertentu.

Jadilah spesifik tentang seperti apa tampan

Semakin konkret dokumen Anda menggambarkan perilaku yang diharapkan, semakin baik Agen AWS Keamanan dapat menentukan kriteria kepatuhan. Di mana Anda bisa, jelaskan apa yang

ditunjukkan oleh desain yang sesuai dan apa yang mengindikasikan pelanggaran. Pernyataan yang tidak jelas atau murni aspirasional menghasilkan persyaratan yang lebih sedikit dan lebih lemah daripada yang konkret dan relevan dengan beban kerja.

Tinjau apa yang Anda dapatkan kembali

Setiap persyaratan yang dihasilkan mencakup nama, penerapan, kriteria kepatuhan, dan panduan remediasi yang diperoleh Agen AWS Keamanan dari dokumen Anda. Tinjau persyaratan yang dihasilkan, edit semua yang perlu disempurnakan, dan aktifkan persyaratan yang ingin Anda evaluasi oleh Agen AWS Keamanan. Untuk informasi selengkapnya, lihat Mengelola persyaratan keamanan.

Kelola pengguna dan akses

Kontrol siapa yang dapat mengakses setiap Ruang Agen, dan konfigurasikan peran IAM dan kunci enkripsi yang digunakan AWS Security Agent.

Berikan pengguna akses ke aplikasi web AWS Security Agent

AWS Security Agent menyediakan dua metode bagi pengguna untuk mengakses aplikasi web, tergantung pada cara Anda mengonfigurasi Ruang Agen selama penyiapan.

Ikhtisar metode akses

IAM Identity Center (SSO) - Jika Anda mengaktifkan IAM Identity Center saat membuat Ruang Agen Anda, pengguna dapat mengakses aplikasi web secara langsung melalui SSO. Anda menetapkan pengguna ke Ruang Agen melalui konsol AWS Security Agent, dan pengguna masuk menggunakan kredensi Pusat Identitas mereka.

Akses Admin - Pengguna dengan akses AWS Console dapat meluncurkan aplikasi web melalui tautan akses admin di halaman ikhtisar Ruang Agen di AWS Management Console.

Akses hibah dengan IAM Identity Center (SSO)

Jika Anda mengonfigurasi Ruang Agen dengan Pusat Identitas IAM, Anda dapat menetapkan pengguna ke Ruang Agen menggunakan konsol AWS Security Agent.

Tetapkan pengguna melalui konsol AWS Security Agent

1. Di AWS Security Agent Management Console, navigasikan ke Ruang Agen Anda.

2. Pilih tab Aplikasi Web.
3. Di tabel Pengguna, pilih Tambahkan pengguna.
4. Pilih pengguna yang ada dari IAM Identity Center atau buat pengguna baru.
5. Konfirmasikan tugas pengguna.

> **Tip**
>
> Pengguna yang ditugaskan ke Ruang Agen dapat mengakses aplikasi web dengan masuk melalui Pusat Identitas IAM dengan kredensi SSO mereka.

Akses aplikasi web dengan SSO

Setelah pengguna ditugaskan ke Ruang Agen:

1. Pengguna menavigasi ke URL aplikasi web untuk Ruang Agen.

   > **Tip**
   >
   > Temukan URL aplikasi web di halaman detail Ruang Agen di konsol AWS Security Agent dengan memilih Salin URL aplikasi web. Pengguna harus menandai URL ini untuk akses mudah.

2. Pengguna masuk menggunakan kredensi SSO mereka.
3. Setelah otentikasi, pengguna dapat memilih Ruang Agen dan mulai melakukan penilaian keamanan.

Berikan akses dengan IAM-only akses

Jika Anda mengonfigurasi Ruang Agen dengan IAM-only akses, pengguna dengan akses AWS Console dapat meluncurkan aplikasi web melalui tautan akses admin.

Gunakan tautan akses admin

1. Masuk ke konsol AWS Security Agent.
2. Arahkan ke Ruang Agen yang ingin Anda akses.

3. Pada tab Aplikasi Web di halaman landing Agent Space pilih tombol akses Admin untuk meluncurkan aplikasi web.
4. Aplikasi web terbuka di tab baru dengan pengguna diautentikasi secara otomatis.

> **Note**
>
> Tautan akses admin hanya tersedia untuk pengguna yang sudah diautentikasi ke AWS Console dengan izin AWS Security Agent yang sesuai. Metode ini tidak memerlukan konfigurasi IAM Identity Center.

Langkah selanjutnya

Setelah memberikan pengguna akses ke aplikasi web:

- Pengguna dapat membuat dan mengelola konfigurasi dan menjalankan pengujian penetrasi
- Pengguna dapat membuat dan mengelola ulasan desain
- Pengguna dapat melihat temuan keamanan dan panduan remediasi
- Konfigurasikan preferensi notifikasi untuk temuan keamanan

Membuat Peran IAM untuk AWS Security Agent

AWS Security Agent menggunakan Peran IAM dalam tiga cara:

1. Peran Aplikasi: Digunakan saat membuat aplikasi AWS Security Agent. Untuk kasus penggunaan IAM Identity Center dan tautan akses admin, layanan mengasumsikan peran ini untuk memberikan izin kepada WebApp pengguna untuk berinteraksi dengan AWS Security Agent API.
2. Peran Layanan Uji Penetrasi: Ditentukan saat membuat spasi agen sebagai daftar peran yang tersedia. Kemudian, WebApp pengguna memilih salah satu peran ini saat membuat tes penetrasi. Layanan AWS Security Agent mengasumsikan peran ini untuk mengakses sumber daya AWS Anda selama pengujian.
3. Peran Aktor: Digunakan untuk mengautentikasi dan mengotorisasi permintaan ke aplikasi web target Anda (misalnya, AWS API Gateway API). Peran ini disediakan selama pembuatan ruang agen. Agen AWS Security Agent mengasumsikan peran aktor untuk berinteraksi dengan aplikasi target Anda.

Peran Aplikasi

Peran Aplikasi digunakan saat membuat aplikasi AWS Security Agent Anda di layanan. Untuk skenario autentikasi Pusat Identitas IAM dan tautan akses admin, layanan AWS Security Agent mengasumsikan peran ini untuk memberikan WebApp pengguna izin yang diperlukan untuk berinteraksi dengan AWS Security Agent API.

Izin yang Diperlukan

Peran ini membutuhkan izin untuk:

- Memanggil operasi AWS Security Agent API
- Membaca dan menulis data konfigurasi aplikasi
- Akses informasi sesi pengguna
- Kelola token otentikasi untuk pengguna WebApp

Kebijakan Kepercayaan

Kebijakan kepercayaan harus mengizinkan layanan AWS Security Agent untuk mengambil peran ini:

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

Kebijakan Izin

Peran harus mencakup izin untuk:

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "securityagent:GetApplication",
        "securityagent:UpdateApplication",
        "securityagent:ListAgentInstances",
        "securityagent:CreatePentestSession"
      ],
      "Resource": "arn:aws:securityagent:*:*:application/*"
    }
  ]
}
```

> **Note**
>
> Sesuaikan izin berdasarkan persyaratan aplikasi spesifik Anda dan prinsip hak istimewa paling sedikit.

Peran Layanan Uji Penetrasi

Peran Layanan Uji Penetrasi ditentukan saat membuat spasi agen sebagai daftar peran yang tersedia. Saat WebApp pengguna membuat tes penetrasi, mereka memilih salah satu peran ini. Layanan AWS Security Agent kemudian mengasumsikan peran ini untuk mengakses dan menguji sumber daya AWS Anda.

Izin yang Diperlukan

Peran ini memerlukan izin untuk mengakses dan menganalisis sumber daya AWS Anda selama pengujian penetrasi:

- Baca dan jelaskan konfigurasi VPC dan topologi jaringan
- Memeriksa instans EC2, grup keamanan, dan ACL jaringan
- Menganalisis kebijakan IAM dan izin sumber daya
- Baca CloudWatch log dan metrik
- Akses konfigurasi layanan AWS yang relevan dengan pengujian keamanan

Kebijakan Kepercayaan

Kebijakan kepercayaan harus memungkinkan layanan AWS Security Agent untuk mengambil peran ini untuk operasi pengujian penetrasi:

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

> **Note**
>
> Gunakan ID Eksternal untuk keamanan tambahan saat mengizinkan akses lintas akun atau layanan.

Kebijakan Izin

Peran tersebut harus mencakup akses hanya-baca ke sumber daya AWS Anda. Pertimbangkan untuk menggunakan kebijakan terkelola ini:

- `SecurityAudit`- Kebijakan terkelola AWS untuk audit keamanan
- `ViewOnlyAccess`- Read-only akses ke sebagian besar layanan AWS

Atau buat kebijakan khusus dengan izin tertentu:

```
{
```

```
  "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

Berikan hanya izin minimum yang diperlukan untuk cakupan pengujian penetrasi Anda. Tinjau dan sesuaikan izin berdasarkan layanan AWS mana yang ingin Anda sertakan dalam pengujian keamanan.

Peran Aktor

Peran Aktor digunakan untuk mengautentikasi dan mengotorisasi permintaan ke aplikasi web target Anda selama pengujian penetrasi. Peran ini disediakan selama pembuatan ruang agen, dan agen Agen Keamanan AWS mengasumsikan peran tersebut berinteraksi dengan titik akhir aplikasi target Anda (seperti API AWS API Gateway, URL fungsi Lambda, atau aplikasi lainnya). AWS-hosted

Izin yang Diperlukan

Peran ini membutuhkan izin untuk:

- Memanggil titik akhir API Gateway

- Jalankan fungsi Lambda
- Akses sumber daya AWS khusus aplikasi
- Mengautentikasi dengan mekanisme otentikasi aplikasi target Anda
- Melakukan operasi HTTP terhadap endpoint aplikasi Anda

Kebijakan Kepercayaan

Kebijakan kepercayaan harus mengizinkan layanan agen AWS Security Agent untuk mengambil peran ini:

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

Kebijakan Izin

Izin bergantung pada arsitektur aplikasi target Anda. Berikut adalah contoh untuk skenario umum:

Untuk Aplikasi API Gateway

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

```
      ],
      "Resource": "arn:aws:execute-api:us-east-1:*:your-api-id/*"
    }
  ]
}
```

Untuk URL Fungsi Lambda

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunctionUrl",
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:us-east-1:*:function:your-function-name"
    }
  ]
}
```

Untuk Application Load Balancer dengan Otentikasi Cognito

```
{
  "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

Konfigurasikan izin Peran Aktor agar sesuai dengan persyaratan autentikasi dan otorisasi aplikasi target Anda. Peran tersebut harus memiliki tingkat akses yang sama dengan

pengguna atau layanan yang akan disimulasikan oleh Agen Keamanan selama pengujian penetrasi.

Kunci terkelola pelanggan untuk AWS Security Agent

Secara default, Agen AWS Keamanan mengenkripsi semua data saat istirahat menggunakan kunci enkripsi AWS-managed. Anda dapat secara opsional menggunakan kunci yang dikelola pelanggan dari AWS Key Management Service (AWS KMS) untuk mengenkripsi data Anda, memberi Anda kontrol penuh atas kunci enkripsi yang melindungi sumber daya Anda.

AWS Agen Keamanan mendukung kunci terkelola pelanggan tingkat sumber daya. Saat Anda membuat sumber daya tingkat atas seperti Ruang Agen atau integrasi, Anda dapat menentukan kunci KMS untuk mengenkripsi semua data milik sumber daya tersebut dan subsumber dayanya. Misalnya, menentukan kunci yang dikelola pelanggan saat membuat Ruang Agen mengenkripsi semua data yang terkait dengan Ruang Agen tersebut, termasuk konfigurasi Ruang Agen, konfigurasi pengujian penetrasi, pekerjaan, dan detail eksekusi, temuan keamanan, titik akhir yang ditemukan, dan tangkapan layar. Anda juga dapat menyediakan AWS sumber daya yang sudah dienkripsi dengan kunci yang dikelola pelanggan Anda sendiri, seperti bucket S3, grup log CloudWatch log, atau rahasia Secrets Manager. Untuk izin KMS yang diperlukan, lihat. the section called “Izin KMS yang diperlukan”

Cara kerja kunci yang dikelola pelanggan

AWS Agen Keamanan menggunakan keyring Hierarki AWS KMS untuk mengenkripsi data Anda dengan kunci yang dikelola pelanggan. Saat Anda menentukan kunci KMS selama pembuatan sumber daya, layanan akan membuat kunci cabang yang dilindungi oleh kunci KMS Anda dan menyimpannya di toko DynamoDB-based kunci Amazon. Untuk setiap operasi enkripsi, keyring Hierarkis memperoleh kunci pembungkus unik dari kunci cabang aktif, yang mengenkripsi kunci enkripsi data unik untuk setiap permintaan.

Gantungan kunci Hierarkis memberikan manfaat berikut:

- Per-resource lingkup enkripsi — Setiap sumber daya tingkat atas memiliki kunci cabangnya sendiri, sehingga data untuk sumber daya yang berbeda dienkripsi di bawah kunci terpisah.
- Rotasi kunci otomatis - Agen AWS Keamanan secara berkala memutar kunci cabang. Setelah rotasi, data baru dienkripsi dengan versi kunci cabang baru, sementara versi yang lebih lama dipertahankan untuk mendekripsi data yang dienkripsi sebelumnya.

Konteks enkripsi

AWS Agen Keamanan menggunakan konteks enkripsi di semua operasi kriptografi dengan kunci KMS Anda. Konteks enkripsi adalah sekumpulan pasangan nilai kunci non-rahasia yang menyediakan data otentikasi tambahan untuk operasi enkripsi.

Kunci konteks enkripsi mengikuti format`aws:securityagent:_<resource-type>_`, di *`<resource-type>`* mana jenis sumber daya yang dienkripsi (misalnya, `agent-space` atau`integration`). Nilainya adalah Nama Sumber Daya Amazon (ARN) dari sumber daya.

> **Note**
>
> Untuk integrasi, kunci konteks enkripsi menyertakan `aws-crypto-ec:` awalan, menghasilkan format. `aws-crypto-ec:aws:securityagent:integration`

Anda dapat menggunakan konteks enkripsi untuk meringkas kebijakan kunci KMS Anda ke jenis sumber daya tertentu. Untuk informasi selengkapnya, lihat the section called “Izin KMS yang diperlukan”.

Enkripsi default

Saat Anda membuat sumber daya tanpa menentukan kunci yang dikelola pelanggan, Agen AWS Keamanan mengenkripsi sumber daya menggunakan kunci enkripsi AWS terkelola yang disediakan oleh layanan penyimpanan yang mendasarinya (Amazon DynamoDB dan Amazon S3). Tidak diperlukan konfigurasi tambahan untuk enkripsi default.

Kunci KMS default untuk aplikasi

Anda dapat mengatur kunci KMS default di tingkat aplikasi. Ketika kunci KMS default dikonfigurasi, Agen AWS Keamanan menggunakannya sebagai fallback untuk Ruang Agen baru dan integrasi ketika Anda tidak secara eksplisit menentukan kunci KMS selama pembuatan sumber daya.

Untuk mengatur kunci KMS default, tentukan `defaultKmsKeyId` parameter saat membuat atau memperbarui aplikasi Anda menggunakan AWS CLI atau SDK. Konsol meminta Anda untuk menentukan kunci KMS default selama pengaturan aplikasi.

Saat Anda secara eksplisit menentukan kunci KMS selama pembuatan sumber daya, kunci tersebut lebih diutamakan daripada default tingkat aplikasi.

Prasyarat

Sebelum Anda mengonfigurasi kunci terkelola pelanggan, lengkapi prasyarat berikut:

- Buat kunci KMS enkripsi simetris di AWS KMS. Kuncinya harus memenuhi persyaratan berikut:
  - Jenis kunci: Simetris
  - Penggunaan kunci: Enkripsi dan dekripsi
  - Spesifikasi kunci: `SYMMETRIC_DEFAULT`
  - Status kunci: Diaktifkan

  Untuk petunjuknya, lihat Membuat AWS kunci di Panduan Pengembang Layanan Manajemen Kunci.
- Konfigurasikan kebijakan kunci KMS dan kebijakan IAM untuk memberikan izin yang diperlukan kepada Agen AWS Keamanan. Lihat perinciannya di the section called “Izin KMS yang diperlukan”.

Izin KMS yang diperlukan

AWS Agen Keamanan memerlukan izin KMS dalam dua skenario:

- Mengenkripsi data dalam Agen AWS Keamanan — Saat Anda menentukan kunci yang dikelola pelanggan untuk Ruang Agen atau integrasi, layanan memerlukan izin untuk menggunakan kunci tersebut untuk operasi kriptografi pada data Anda.
- Menggunakan CMK-encrypted AWS sumber daya — Saat Anda menyediakan AWS sumber daya yang dienkripsi dengan kunci yang dikelola pelanggan (seperti bucket S3, grup log CloudWatch log, atau rahasia Secrets Manager), layanan memerlukan izin untuk menggunakan kunci tersebut untuk mengakses sumber daya terenkripsi.

AWS Agen Keamanan mengakses kunci KMS Anda menggunakan identitas yang berbeda tergantung pada operasi:

- AWS Operasi Konsol Manajemen — Saat administrator membuat atau mengelola sumber daya di Konsol AWS Manajemen (misalnya, membuat Ruang Agen atau memperbarui aplikasi), layanan menggunakan peran administrator untuk memanggil AWS KMS.
- Operasi aplikasi web — Ketika pengguna IAM Identity Center mengakses data melalui aplikasi web Agen AWS Keamanan (misalnya, melihat hasil tes penetrasi atau memulai tes penetrasi),

layanan menggunakan peran aplikasi yang dibuat selama penyiapan Agen AWS Keamanan untuk memanggil AWS KMS.

- Mengakses sumber daya yang disediakan pelanggan — Saat layanan mengakses sumber AWS daya yang disediakan pelanggan selama pengujian penetrasi (seperti bucket S3, CloudWatch grup log log, dan rahasia Secrets Manager), layanan ini menggunakan peran layanan pengujian penetrasi untuk memanggil KMS. AWS
- Alur kerja asinkron — Untuk operasi latar belakang yang berjalan di luar sesi pengguna (misalnya, eksekusi pengujian penetrasi, pemrosesan peninjauan kode, dan rotasi kunci cabang), layanan menggunakan prinsipal layanannya sendiri (`securityagent.amazonaws.com`) untuk memanggil AWS KMS atas nama Anda.

Bagian berikut menjelaskan izin yang diperlukan untuk setiap identitas.

Kebijakan kunci

Kebijakan kunci KMS Anda harus memberikan izin Agen AWS Keamanan untuk menggunakan kunci untuk operasi kriptografi. Pernyataan kebijakan kunci yang diperlukan bergantung pada jenis sumber daya yang Anda rencanakan untuk dienkripsi dan CMK-encrypted AWS sumber daya yang Anda berikan ke layanan.

Kebijakan utama untuk Agen Spaces

Kebijakan utama berikut memberi Agen AWS Keamanan izin yang diperlukan untuk mengenkripsi dan mendekripsi data Ruang Agen, termasuk hasil pengujian penetrasi dan tangkapan layar.

Ganti nilai placeholder berikut dalam kebijakan:

- *111122223333* — ID AWS akun Anda
- *MyRole* — Peran IAM yang Anda gunakan untuk mengelola Agen AWS Keamanan di konsol
- *MyApplicationRole* — Peran aplikasi yang dibuat selama penyiapan Agen AWS Keamanan
- *us-east-1* — AWS Wilayah tempat Anda menggunakan Agen AWS Keamanan

```
{
  "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**
>
> Untuk memutar kunci cabang, kebijakan kunci KMS Anda harus memberikan `kms:GenerateDataKeyWithoutPlaintextkms:ReEncryptTo`, dan `kms:ReEncryptFrom` izin ke prinsipal layanan Agen AWS Keamanan (`securityagent.amazonaws.com`), atau rotasi kunci cabang akan gagal. Untuk informasi selengkapnya tentang izin yang diperlukan, lihat Memutar kunci cabang.

Kebijakan utama untuk integrasi

Kebijakan utama berikut memberi Agen AWS Keamanan izin yang diperlukan untuk mengenkripsi dan mendekripsi data integrasi, termasuk temuan tinjauan kode.

Ganti nilai placeholder berikut dalam kebijakan:

- *111122223333* — ID AWS akun Anda
- *MyRole* — Peran IAM yang Anda gunakan untuk mengelola Agen AWS Keamanan di konsol
- *us-east-1* — AWS Wilayah tempat Anda menggunakan Agen AWS Keamanan

```
{
  "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**
>
> Anda dapat menggabungkan pernyataan kebijakan kunci paket persyaratan Agen, integrasi, dan paket persyaratan keamanan ke dalam kebijakan kunci tunggal jika Anda ingin menggunakan kunci KMS yang sama untuk beberapa jenis sumber daya.

Kebijakan utama untuk paket persyaratan keamanan

Kebijakan utama berikut memberi Agen AWS Keamanan izin yang diperlukan untuk mengenkripsi dan mendekripsi data paket persyaratan keamanan. Paket persyaratan keamanan tidak menggunakan peran aplikasi web. Kebijakan ini menggunakan dua jalur akses:

- Jalur sinkron — Saat Anda memanggil API, layanan menggunakan kredensi yang diteruskan untuk memanggil AWS KMS atas nama Anda (melalui kondisi). `kms:ViaService`
- Jalur asinkron — Untuk operasi asinkron di mana paket dapat digunakan, layanan menggunakan prinsipal layanannya sendiri untuk memanggil KMS secara langsung. AWS

Ganti nilai berikut dalam kebijakan:

- *`111122223333`* — ID AWS akun Anda
- *`MyRole`* — Peran IAM yang Anda gunakan untuk menjalankan operasi paket persyaratan keamanan
- *`us-east-1`* — AWS Wilayah tempat Anda menggunakan Agen AWS Keamanan

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Kebijakan tersebut berisi tiga pernyataan:

- AllowKeyMetadataValidation— Memberikan `kms:DescribeKey` agar Agen AWS Keamanan dapat memvalidasi bahwa kunci yang dikelola pelanggan Anda ada, diaktifkan, dan menggunakan enkripsi simetris saat Anda menentukan CMK selama pembuatan sumber daya. Menggunakan `kms:ViaService` kondisi untuk memastikan panggilan berasal melalui layanan.
- AllowUseOfHierarchicalKeyringForSecurityPacks— Hibah `kms:GenerateDataKeyWithoutPlaintext` (buat kunci cabang baru), `kms:ReEncryptTo` dan `kms:ReEncryptFrom` (putar kunci cabang yang ada), dan `kms:Decrypt` (ambil kunci cabang yang ada) untuk operasi keyring hierarkis. Operasi ini menggunakan kredensil yang diteruskan melalui jalur sinkron dan dicakup ke sumber daya paket persyaratan keamanan dengan kondisi konteks enkripsi.
- AllowAsynchronousDataAccessForSecurityPacks— Memberikan`kms:GenerateDataKeyWithoutPlaintext`,, `kms:Decrypt`kms:ReEncryptTo, dan `kms:ReEncryptFrom` kepada prinsipal layanan Agen AWS Keamanan untuk operasi asinkron ketika tidak ada kredensyal pemanggil yang tersedia.

Berbeda dengan kebijakan kunci Agent Spaces, kebijakan paket persyaratan keamanan tidak menyertakan prinsipal peran aplikasi web. Paket persyaratan keamanan diakses melalui Konsol AWS Manajemen menggunakan peran administrator Anda, bukan melalui aplikasi web Agen AWS Keamanan. Semua operasi sinkron menggunakan peran pemanggil tunggal (via`kms:ViaService`).

> **Note**
>
> Jika Anda menggunakan kunci KMS yang sama untuk Ruang Agen dan paket persyaratan keamanan, Anda dapat menggabungkan pernyataan kebijakan utama dari kedua bagian ke dalam satu kebijakan kunci.

Kebijakan utama untuk CMK-encrypted AWS sumber daya

Jika Anda menyediakan AWS sumber daya yang dienkripsi dengan kunci terkelola pelanggan ke Agen AWS Keamanan, Anda harus memperbarui kebijakan kunci pada setiap kunci KMS sumber daya untuk memberikan izin yang diperlukan. Ini berlaku untuk sumber daya berikut:

- Bucket S3 — Jika Anda menyediakan sumber belajar dari bucket S3 yang dienkripsi dengan kunci terkelola pelanggan (SSE-KMS), kebijakan kunci harus mengizinkan peran layanan pengujian penetrasi untuk mendekripsi objek.
- Rahasia Secrets Manager — Jika kredensyal pengujian penetrasi Anda disimpan dalam rahasia Secrets Manager yang dienkripsi dengan kunci yang dikelola pelanggan, kebijakan kunci harus mengizinkan peran layanan pengujian penetrasi untuk mendekripsi rahasia tersebut. Jika aplikasi web membuat rahasia atas nama Anda, kebijakan kunci juga harus memungkinkan peran aplikasi untuk mengenkripsi rahasia tersebut.
- CloudWatch Grup log log - Jika grup CloudWatch log Log yang digunakan untuk menyimpan log eksekusi uji penetrasi dienkripsi dengan kunci yang dikelola pelanggan, kebijakan kunci harus mengizinkan CloudWatch Log untuk memvalidasi kunci KMS melalui peran layanan dan memungkinkan prinsipal layanan CloudWatch Log untuk melakukan operasi kriptografi.

> **Important**
>
> AWS Agen Keamanan juga dapat membuat grup CloudWatch log Log dan rahasia Secrets Manager atas nama Anda. Saat mengonfigurasi kebijakan kunci KMS Anda, sertakan juga pernyataan terkait untuk sumber daya yang dibuat layanan ini.

Pernyataan kebijakan utama berikut memberikan izin yang diperlukan untuk CMK-encrypted sumber daya. Sertakan hanya pernyataan yang berlaku untuk konfigurasi Anda. Jika sumber daya menggunakan kunci KMS yang sama yang Anda tentukan untuk Ruang Agen Anda, tambahkan

pernyataan ini ke kebijakan kunci tersebut. Jika sumber daya menggunakan kunci KMS yang berbeda, tambahkan pernyataan ini ke kebijakan kunci tersebut.

Ganti nilai placeholder berikut dalam kebijakan:

- *111122223333* — ID AWS akun Anda
- *MyApplicationRole* — Peran aplikasi yang dibuat selama penyiapan Agen AWS Keamanan
- *MyPenTestServiceRole* — Peran layanan uji penetrasi
- *us-east-1* — AWS Wilayah tempat Anda menggunakan Agen AWS Keamanan

```
{
  "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`Pernyataan ini diperlukan hanya jika Anda menyediakan sumber belajar dari bucket S3 yang dienkripsi dengan kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang mengonfigurasi SSE-KMS S3, lihat Melindungi data dengan. SSE-KMS

- `AllowKmsKeyDecryptionForSecretsManagerSecrets`Pernyataan ini diperlukan hanya jika kredenal pengujian penetrasi Anda disimpan dalam rahasia Secrets Manager yang dienkripsi dengan kunci yang dikelola pelanggan. Peran layanan membutuhkan akses untuk mendekripsi rahasia selama pengujian penetrasi. Untuk informasi selengkapnya tentang enkripsi rahasia, lihat Enkripsi dan dekripsi rahasia di Secrets Manager.
- `AllowKmsKeyValidationForCloudWatchLogsLogGroups`Pernyataan memberikan peran layanan `kms:DescribeKey` sehingga CloudWatch Log dapat memvalidasi kunci KMS melalui peran layanan saat mengaitkannya dengan grup log.
- `AllowKmsKeyAccessForCreatingSecrets`Pernyataan ini diperlukan jika aplikasi web membuat rahasia Secrets Manager atas nama Anda menggunakan kunci yang dikelola pelanggan. Peran aplikasi `kms:Decrypt` perlu `kms:GenerateDataKey` dan mengenkripsi rahasia dengan kunci KMS Anda.
- `AllowKmsKeyAccessForCloudWatchLogsLogGroups`Pernyataan tersebut memberikan prinsip layanan CloudWatch Logs (`logs.`*`us-east-1`*`.amazonaws.com`) izin yang diperlukan untuk mengenkripsi dan mendekripsi data log. Pernyataan ini juga diperlukan jika Agen AWS Keamanan membuat grup log atas nama Anda saat Anda tidak menyediakan grup yang sudah ada. Untuk informasi selengkapnya, lihat Mengenkripsi data log di CloudWatch Log menggunakan AWS KMS.
- Jika beberapa sumber daya berbagi kunci KMS yang sama, Anda dapat menggabungkan pernyataan ke dalam satu kebijakan kunci.

Peran administrator

Kebijakan IAM tambahan tidak diperlukan untuk peran administrator (peran IAM yang Anda gunakan untuk mengelola Agen AWS Keamanan di konsol).

Peran aplikasi

Selain kebijakan kunci KMS, lampirkan kebijakan IAM berikut ke peran aplikasi yang ditentukan selama penyiapan Agen AWS Keamanan. Kebijakan ini memberikan izin peran untuk menggunakan kunci terkelola pelanggan Anda untuk mengenkripsi dan mendekripsi data Ruang Agen dan membuat rahasia di Secrets Manager. AWS

Ganti nilai placeholder berikut dalam kebijakan:

- *111122223333* — ID AWS akun Anda

- *us-east-1* — AWS Wilayah tempat Anda menggunakan Agen AWS Keamanan
- ARN kunci KMS di Resource — ARN kunci KMS Anda

```
{
  "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**
>
> Peran aplikasi adalah peran IAM yang Anda tentukan atau yang dibuat konsol selama penyiapan Agen AWS Keamanan. Aplikasi web mengasumsikan peran ini untuk mengambil log eksekusi uji penetrasi dan membuat rahasia Secrets Manager atas nama Anda.
>
> - `AllowKmsKeyAccessForCreatingSecrets`Pernyataan ini diperlukan jika Anda mengonfigurasi sumber daya otentikasi untuk pengujian penetrasi dan memilih untuk memasukkan kredensyal secara langsung alih-alih menentukan rahasia yang ada. Aplikasi web membuat rahasia atas nama Anda, dan peran aplikasi memerlukan izin dalam pernyataan ini untuk mengenkripsi rahasia dengan kunci terkelola pelanggan yang ditentukan untuk Ruang Agen Anda. Perbarui `Resource` ARN dalam pernyataan ini agar sesuai dengan kunci KMS yang digunakan untuk Ruang Agen Anda.

Peran layanan

Selama pengujian penetrasi, Agen AWS Keamanan mengasumsikan peran layanan uji penetrasi untuk mengakses AWS sumber daya Anda. Jika salah satu sumber daya ini dienkripsi dengan kunci yang dikelola pelanggan, Anda harus memberikan izin peran layanan untuk menggunakan kunci KMS yang sesuai. Ini berlaku untuk sumber daya berikut:

- Bucket S3 — Jika Anda menyediakan sumber daya pembelajaran (seperti dokumen API, model ancaman, atau kode sumber) dari bucket S3 yang dienkripsi dengan kunci yang dikelola pelanggan, peran layanan memerlukan izin untuk mendekripsi objek dalam bucket tersebut. Untuk informasi selengkapnya tentang mengonfigurasi SSE-KMS S3, lihat Melindungi data dengan. SSE-KMS

- Rahasia Secrets Manager — Jika kredensyal pengujian penetrasi Anda disimpan dalam rahasia Secrets Manager yang dienkripsi dengan kunci yang dikelola pelanggan, peran layanan memerlukan izin untuk mendekripsi rahasia tersebut. Untuk informasi selengkapnya tentang enkripsi rahasia, lihat Enkripsi dan dekripsi rahasia di Secrets Manager.
- CloudWatch Grup log log - Jika grup CloudWatch log Log yang digunakan untuk menyimpan log eksekusi uji penetrasi dienkripsi dengan kunci yang dikelola pelanggan (termasuk grup log yang dibuat oleh Agen AWS Keamanan atas nama Anda), peran layanan memerlukan `kms:DescribeKey` izin untuk memvalidasi kunci. Prinsipal layanan CloudWatch Log menangani enkripsi dan dekripsi data log yang sebenarnya, dan izin tersebut diberikan melalui kebijakan kunci (lihat). the section called “Kebijakan kunci” Untuk informasi selengkapnya tentang mengenkripsi data log, lihat Mengenkripsi data CloudWatch log di Log menggunakan KMS. AWS

> **Important**
>
> Jika Anda menggunakan kunci KMS yang berbeda untuk mengenkripsi sumber daya ini daripada yang Anda tentukan untuk Ruang Agen, Anda harus memberikan izin peran layanan untuk setiap kunci KMS yang melindungi sumber daya yang diakses peran layanan selama pengujian penetrasi.

Lampirkan kebijakan IAM berikut ke peran layanan pengujian penetrasi. Sertakan hanya pernyataan yang berlaku untuk konfigurasi Anda.

Ganti nilai placeholder berikut dalam kebijakan:

- *111122223333* — ID AWS akun Anda
- *us-east-1* — AWS Wilayah tempat Anda menggunakan Agen AWS Keamanan
- ARN kunci KMS di `Resource` — ARN dari kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi setiap sumber daya

```
{
  "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`Pernyataan ini diperlukan hanya jika Anda menyediakan sumber belajar dari bucket S3 yang dienkripsi dengan kunci yang dikelola pelanggan. Perbarui `Resource` ARN agar sesuai dengan kunci KMS yang digunakan untuk mengenkripsi bucket S3 Anda, dan perbarui `kms:EncryptionContext:aws:s3:arn` nilainya agar sesuai dengan nama bucket Anda.
- `AllowKmsAccessForEncryptedSecrets`Pernyataan ini diperlukan hanya jika kredenal pengujian penetrasi Anda disimpan dalam rahasia Secrets Manager yang dienkripsi dengan kunci yang dikelola pelanggan. Perbarui `Resource` ARN agar sesuai dengan kunci KMS yang digunakan untuk mengenkripsi rahasia Anda, dan perbarui `kms:EncryptionContext:SecretARN` nilainya agar sesuai dengan nama rahasia Anda.
- `AllowKmsKeyValidationForCloudWatchLogsLogGroups`Pernyataan hanya diperlukan jika grup CloudWatch log Log Anda dienkripsi dengan kunci yang dikelola pelanggan, termasuk grup log yang dibuat oleh Agen AWS Keamanan atas nama Anda. Perbarui `Resource` ARN agar sesuai dengan kunci KMS yang digunakan untuk mengenkripsi grup log Anda.
- Jika beberapa sumber daya berbagi kunci KMS yang sama, Anda dapat menggabungkan pernyataan dan mencantumkan ARN kunci bersama sekali di `Resource` bidang.

Buat sumber daya dengan kunci yang dikelola pelanggan

Anda dapat menentukan kunci yang dikelola pelanggan saat menyiapkan Agen AWS Keamanan atau saat membuat sumber daya individual. Kunci KMS mengenkripsi semua data milik sumber daya itu dan subsumber dayanya.

Tetapkan kunci KMS default selama penyiapan (konsol)

Anda dapat mengonfigurasi kunci KMS default selama penyiapan Agen AWS Keamanan awal. Kunci ini digunakan sebagai default untuk Ruang Agen baru dan integrasi kecuali Anda menentukan kunci yang berbeda.

1. Pada halaman Siapkan AWS Security Agent, perluas bagian Enkripsi - opsional.
2. Pilih Sesuaikan pengaturan enkripsi (lanjutan).
3. Untuk Pilih kunci AWS KMS, pilih kunci KMS dari akun Anda, atau masukkan ARN kunci KMS.
4. Selesaikan langkah-langkah penyiapan yang tersisa dan pilih Siapkan AWS Security Agent.

AWS Agen Keamanan memvalidasi kunci KMS untuk mengonfirmasi bahwa kunci itu ada, diaktifkan, dan menggunakan enkripsi simetris. Jika validasi gagal, penyiapan tidak dilanjutkan dan Anda menerima pesan kesalahan.

Membuat Ruang Agen dengan kunci yang dikelola pelanggan (konsol)

1. Di konsol Agen AWS Keamanan, navigasikan ke halaman Agen Spaces.
2. Pilih Buat Ruang Agen.
3. Masukkan nama dan deskripsi opsional untuk Ruang Agen Anda.
4. Perluas bagian Advanced.
5. Di bawah Enkripsi data, pilih salah satu dari berikut ini:
   - Gunakan kunci default aplikasi - Menggunakan kunci KMS default yang dikonfigurasi selama pengaturan Agen AWS Keamanan. Opsi ini dipilih secara default jika kunci default dikonfigurasi.
   - Gunakan tombol yang berbeda - Tentukan kunci KMS yang berbeda untuk Ruang Agen ini. Pilih Sesuaikan pengaturan enkripsi (lanjutan), lalu untuk Pilih kunci AWS KMS, pilih kunci KMS dari akun Anda atau masukkan ARN.
6. Pilih Buat.

Buat integrasi dengan kunci yang dikelola pelanggan (konsol)

1. Di konsol Agen AWS Keamanan, navigasikan ke halaman Integrasi.
2. Pilih Tambahkan integrasi dan pilih penyedia Anda (misalnya, GitHub).
3. Selesaikan Langkah 1: Instal dan otorisasi aplikasi penyedia.
4. Pada Langkah 2: Daftarkan detail, masukkan nama pendaftaran dan konfigurasikan pengaturan penyedia.
5. Di bagian Enkripsi data, pilih Sesuaikan pengaturan enkripsi (lanjutan).
6. Untuk Pilih kunci AWS KMS, pilih kunci KMS dari akun Anda, atau masukkan ARN kunci KMS.
7. Pilih Hubungkan.

Buat sumber daya dengan kunci yang dikelola pelanggan (AWS CLI atau SDK)

Untuk menentukan kunci yang dikelola pelanggan menggunakan AWS CLI atau SDK:

- Sertakan `defaultKmsKeyId` parameter saat memanggil `CreateApplication` untuk menyetel kunci KMS default untuk aplikasi Anda. Kunci ini digunakan sebagai fallback saat membuat Ruang Agen atau integrasi tanpa kunci KMS eksplisit.
- Sertakan `kmsKeyId` parameter saat memanggil `CreateAgentSpace` atau `CreateIntegration` mengenkripsi sumber daya tertentu. Ini lebih diutamakan daripada default tingkat aplikasi.

Untuk detail parameter, lihat Referensi API Agen AWS Keamanan.

Jika Anda tidak menentukan kunci KMS, Agen AWS Keamanan menggunakan kunci KMS default tingkat aplikasi jika salah satu dikonfigurasi. Jika tidak, sumber daya dienkripsi dengan kunci AWS-managed.

Rotasi kunci

AWS Agen Keamanan secara otomatis memutar kunci cabang secara berkala. Rotasi kunci cabang membuat versi aktif baru dari kunci cabang sambil mempertahankan semua versi sebelumnya. Setelah rotasi:

- Data baru dienkripsi dengan versi kunci cabang baru.
- Data yang dienkripsi sebelumnya tetap dapat didekripsi menggunakan versi kunci cabang yang lebih lama.

- Tidak diperlukan enkripsi ulang data.

Rotasi kunci cabang terpisah dari rotasi kunci KMS. Anda dapat mengaktifkan rotasi kunci KMS otomatis untuk kunci yang dikelola pelanggan Anda secara independen. Untuk informasi selengkapnya, lihat Memutar kunci KMS di Panduan Pengembang Layanan Manajemen AWS Kunci.

Setelah rotasi kunci cabang, ada periode singkat di mana salinan cache dari kunci cabang sebelumnya masih dapat digunakan untuk operasi enkripsi baru. Jendela ini ditentukan oleh cache internal time-to-live (TTL) dan tidak mempengaruhi keamanan data Anda.

Pertimbangan-pertimbangan

Ingatlah hal berikut saat menggunakan kunci yang dikelola pelanggan dengan Agen AWS Keamanan:

- Kunci yang dikelola pelanggan hanya berlaku untuk sumber daya baru. Sumber daya yang ada yang dibuat sebelum Anda mengonfigurasi kunci terkelola pelanggan terus menggunakan enkripsi AWS-managed. Tidak ada migrasi data yang ada.
- Kunci yang dikelola pelanggan ditentukan pada waktu pembuatan. Anda menentukan kunci KMS saat membuat sumber daya. Anda tidak dapat menambahkan atau mengubah kunci KMS untuk sumber daya yang ada.
- Beberapa tombol KMS didukung. Anda dapat menggunakan kunci KMS yang berbeda untuk sumber daya yang berbeda. Misalnya, Anda dapat menggunakan satu kunci untuk Ruang Agen produksi dan kunci lain untuk pengembangan Ruang Agen.
- Menonaktifkan atau menghapus kunci KMS membuat data tidak dapat diakses. Jika Anda menonaktifkan atau menjadwalkan penghapusan kunci KMS, Agen AWS Keamanan tidak dapat mendekripsi data yang dienkripsi di bawah kunci tersebut. Sumber daya yang terpengaruh menjadi tidak dapat diakses sampai kunci diaktifkan kembali. Menghapus kunci KMS secara permanen mencegah akses ke semua data yang dienkripsi di bawah kunci itu.
- Izin kebijakan utama diperlukan. Jika Anda menghapus izin yang diperlukan dari kebijakan kunci KMS Anda, Agen AWS Keamanan tidak dapat mengakses data terenkripsi. Pastikan kebijakan utama memberikan izin untuk peran administrator (digunakan untuk mengelola layanan di AWS Konsol), peran aplikasi (digunakan oleh aplikasi web), peran layanan pengujian penetrasi (diasumsikan oleh agen untuk menjalankan pengujian penetrasi), dan prinsip layanan Agen AWS Keamanan seperti yang dijelaskan dalam. the section called “Izin KMS yang diperlukan”
- AWS Kuota KMS berlaku. Operasi kriptografi terhadap kunci KMS Anda dihitung terhadap kuota permintaan AWS KMS Anda. Dalam penggunaan normal, arsitektur keyring hierarkis

meminimalkan panggilan KMS melalui caching kunci cabang. Untuk informasi selengkapnya, lihat Meminta kuota di Panduan Pengembang Layanan Manajemen AWS Kunci.

Pemecahan Masalah

Temukan solusi untuk kesalahan yang umum terlihat saat menggunakan AWS Security Agent.

Akses Ditolak: Jenis GitHub akun salah dipilih atau nama organisasi yang salah ditentukan

- Anda menginstal aplikasi AWS Security Agent ke GitHub organisasi yang Anda inginkan tetapi salah menyetel `GitHub Account Type` ke, `User` bukan `Organization`
- Anda menginstal aplikasi AWS Security Agent ke GitHub organisasi yang Anda inginkan dan `GitHub Account Type` mengaturnya dengan benar `Organization` tetapi membiarkan `Organization Name` bidang kosong atau memasukkan nama organisasi yang salah yang tidak cocok dengan organisasi tempat Anda menginstal aplikasi

Solusi:

1. Buka GitHub dan hapus instalan aplikasi dari organisasi. Untuk informasi selengkapnya, lihat the section called “Langkah 1: Copot pemasangan GitHub Aplikasi AWS Security Agent dari GitHub”.
2. Kembali ke halaman integrasi dan mulai ulang proses integrasi dengan mengklikAdd `Integrations`, menginstal, dan mengotorisasi aplikasi ke GitHub organisasi yang Anda inginkan sekali lagi.
3. Pilih `Organization` dari dropdown `GitHub account type`
4. Pastikan masukan `Organization Name` Anda sama persis dengan yang Anda instal aplikasi.
5. Pilih Connect untuk membuat integrasi GitHub organisasi Anda.

Akses Ditolak: Izin tidak cukup untuk menginstal GitHub Aplikasi ke dalam organisasi

Saat Anda mencoba menginstal aplikasi AWS Security Agent ke GitHub organisasi yang Anda inginkan, Anda akan melihat dua pesan berbeda pada tombol di halaman instalasi.

- Sebuah organisasi `Member` akan melihat `Authorize & Request`

- Sebuah organisasi `Owner` akan melihat `Install & Authorize`

Anda dapat memverifikasi apakah Anda seorang `Member` atau `Owner` dari GitHub organisasi dengan mengikuti langkah-langkah di bawah ini.

1. Pergi ke github.com
2. Pilih profil Anda di situs web
3. Arahkan `Organizations` ke menu dropdown dan pilih
4. Temukan organisasi tempat Anda ingin menginstal AWS Security Agent dari daftar organisasi, yang akan menentukan apakah Anda seorang `Member` atau di `Owner` sebelah nama organisasi.

Solusi yang mungkin:

- Mintalah pemilik menyetujui permintaan instalasi Anda SEBELUM Anda mencoba membuat integrasi
- Mintalah pemilik memperbarui peran Anda dalam GitHub organisasi dari a `Member` ke an `Owner` dan mulai ulang proses integrasi lagi

Agen tidak dapat terhubung ke titik akhir selama tes penetrasi

Jika agen uji penetrasi tidak dapat melakukan panggilan ke URL target yang dikonfigurasi atau gagal menavigasi titik akhir target dengan sukses:

- Jika titik akhir Anda melakukan panggilan ke domain di luar URL target yang dikonfigurasi, verifikasi bahwa domain tambahan ditambahkan sebagai URL yang Dapat Diakses dalam konfigurasi pentest
- Pengujian penetrasi saat ini hanya tersedia untuk HTTP/HTTPS titik akhir yang melayani lalu lintas di port 80 atau 443
- Jika Anda memiliki WAF yang dikonfigurasi, periksa apakah WAF Anda tidak memblokir lalu lintas pengujian penetrasi. Anda dapat mengizinkan lalu lintas uji penetrasi daftar dengan `User-Agent` header, yang akan diatur ke `securityagent` secara default

Mendapatkan bantuan tambahan

Jika Anda terus mengalami masalah setelah mencoba langkah-langkah pemecahan masalah ini:

- Periksa halaman status layanan AWS Security Agent
- Hubungi AWS Support untuk bantuan terkait masalah konfigurasi yang rumit

Untuk pengguna dan pengembang (aplikasi web dan IDE)

Jalankan tinjauan desain, model ancaman, ulasan kode, dan tes penetrasi di aplikasi web, IDE, atau penyedia sumber yang terhubung, lalu tinjau dan pulihkan temuan.

Masuk ke Aplikasi Web AWS Security Agent

Anda harus masuk ke AWS Security Agent Web App untuk menyelesaikan tugas-tugas seperti:

- Lakukan tinjauan desain
- Lakukan tes penetrasi

Metode akses

AWS Security Agent menyediakan berbagai metode untuk mengakses aplikasi web, bergantung pada cara organisasi Anda mengonfigurasi akses selama penyiapan awal dan apakah Anda memiliki akses AWS Console.

IAM Identity Center (SSO) - Jika organisasi Anda mengaktifkan IAM Identity Center, Anda dapat masuk menggunakan kredensi SSO Anda. Anda harus ditugaskan ke setidaknya satu Ruang Agen di Pusat Identitas IAM sebelum Anda dapat mengakses aplikasi web.

Akses admin (IAM) - Jika Anda memiliki akses AWS Console dengan izin yang sesuai, Anda dapat meluncurkan aplikasi web melalui tautan akses admin di halaman Ruang Agen mana pun. Metode ini menyediakan otentikasi melalui sesi Konsol yang ada.

> **Note**
>
> Metode akses utama yang digunakan organisasi Anda ditentukan selama penyiapan AWS Security Agent awal. Untuk informasi tentang mengonfigurasi akses pengguna dan penetapan, lihat. the section called “Berikan pengguna akses ke aplikasi web AWS Security Agent”

Masuk dengan IAM Identity Center (SSO)

Jika organisasi Anda mengonfigurasi akses Pusat Identitas IAM, Anda dapat masuk ke aplikasi web menggunakan kredensil SSO Anda melalui beberapa titik masuk.

Prasyarat

- Anda telah ditugaskan ke setidaknya satu Ruang Agen di Pusat Identitas IAM
- Anda memiliki kredensi IAM Identity Center SSO

Akses aplikasi web

Pilih salah satu metode berikut berdasarkan kebutuhan Anda:

Untuk melihat semua Ruang Agen yang Anda tetapkan:

1. Di konsol AWS Security Agent, navigasikan ke Pengaturan.
2. Temukan domain aplikasi Web dan salin URL.

> **Tip**
>
> Tandai URL ini untuk akses mudah. Ini adalah titik masuk universal untuk melihat semua Ruang Agen yang Anda tetapkan.

3. Arahkan ke URL aplikasi web.
4. Masukkan kredensil Pusat Identitas IAM Anda saat diminta.
5. Setelah otentikasi, Anda akan melihat daftar semua Ruang Agen yang Anda tetapkan.
6. Pilih Ruang Agen tempat Anda ingin bekerja.

Untuk mengakses Ruang Agen tertentu secara langsung (memerlukan akses AWS Console):

1. Masuk ke AWS Management Console.
2. Arahkan ke konsol AWS Security Agent.
3. Arahkan ke Ruang Agen yang ingin Anda akses.
4. Pilih salah satu cara berikut:
   - Luncurkan tombol aplikasi web di halaman ikhtisar Agent Space
   - Luncurkan tombol aplikasi web di bagian Peninjauan kode
   - Luncurkan tombol aplikasi web di bagian pengujian Penetrasi
5. Masukkan kredensil Pusat Identitas IAM Anda saat diminta.
6. Setelah otentikasi, Anda akan dibawa langsung ke Ruang Agen di aplikasi web.

> **Note**
>
> Jika Anda tidak memiliki akses AWS Console, gunakan domain aplikasi web dari halaman Pengaturan untuk mengakses semua Ruang Agen yang ditetapkan.

Masuk dengan akses admin (IAM)

Jika Anda memiliki akses AWS Console dengan izin AWS Security Agent yang sesuai, Anda dapat meluncurkan aplikasi web melalui tautan akses admin dengan autentikasi otomatis.

Prasyarat

- Anda masuk ke AWS Management Console
- Anda memiliki izin yang sesuai untuk mengakses sumber daya AWS Security Agent

Akses aplikasi web

Pilih salah satu dari metode berikut:

Untuk mengakses Ruang Agen tertentu:

1. Masuk ke AWS Management Console.
2. Arahkan ke konsol AWS Security Agent.
3. Arahkan ke Ruang Agen yang ingin Anda akses.
4. Pilih tombol akses Admin di halaman ikhtisar Ruang Agen.
5. Aplikasi web terbuka di tab baru dengan otentikasi otomatis untuk Ruang Agen itu.

> **Note**
>
> Tombol akses admin selalu tersedia untuk pengguna dengan akses AWS Console, terlepas dari apakah organisasi Anda menggunakan IAM Identity Center sebagai metode akses utama.

Buat tinjauan desain

Nilai dokumen desain Anda berdasarkan persyaratan keamanan organisasi dengan mengunggah file untuk ditinjau oleh AWS Security Agent. Tinjauan desain membantu mengidentifikasi masalah keamanan di awal siklus hidup pengembangan, memungkinkan Anda mengatasi masalah arsitektur saat masalah tersebut paling hemat biaya untuk diselesaikan.

AWS Security Agent menganalisis dokumen desain Anda sesuai dengan persyaratan keamanan organisasi Anda, memberikan temuan keamanan terperinci untuk meningkatkan postur keamanan sebelum implementasi dimulai.

Dalam prosedur ini, Anda akan membuat tinjauan desain dengan mengunggah file desain untuk analisis.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Akses ke aplikasi web AWS Security Agent
- Dokumen desain siap diunggah (DOC, DOCX, JPEG, MD, PDF, PNG dan TXT)
- Setiap file harus berukuran 2MB atau lebih kecil, dengan total gabungan 6MB di semua file
- Memahami persyaratan keamanan yang diaktifkan untuk organisasi Anda

Langkah 1: Mulai membuat tinjauan desain

Arahkan ke halaman pembuatan tinjauan desain di Aplikasi Web Agen.

1. Masuk ke aplikasi web AWS Security Agent.
2. Arahkan ke bagian Ulasan desain.
3. Pilih Buat Ulasan Desain.

> **Tip**
>
> Anda dapat melihat persyaratan keamanan yang diaktifkan organisasi Anda dengan menavigasi ke halaman Persyaratan Keamanan di konsol AWS Security Agent. Pilih

persyaratan yang diaktifkan untuk melihat detailnya. Persyaratan ini digunakan untuk menganalisis file desain Anda.

Langkah 2: Beri nama ulasan desain Anda

Berikan nama deskriptif yang membantu mengidentifikasi tujuan dan ruang lingkup tinjauan desain ini.

1. Di bagian Nama tinjauan desain, cari bidang Nama.
2. Masukkan nama deskriptif untuk tinjauan desain Anda.

> **Note**
>
> Nama harus dengan jelas mengidentifikasi proyek, fitur, atau komponen yang sedang ditinjau. Maksimal 80 karakter.

Langkah 3: Unggah file desain

Unggah dokumen desain yang ingin dianalisis oleh AWS Security Agent untuk kepatuhan keamanan.

1. Di bagian File untuk ditinjau, tinjau persyaratan file:

> **Important**
>
> Maksimal 5 file dapat diunggah per tinjauan desain. Setiap file harus berukuran 2MB atau lebih kecil, dengan total gabungan 6MB di semua file. Format yang didukung: DOC, DOCX, JPEG, MD, PDF, PNG dan TXT.

2. Unggah file Anda menggunakan salah satu metode berikut:
   a. Seret dan lepas - Seret file langsung ke area dropzone file
   b. Browse - Gunakan tombol Pilih file untuk menelusuri dan memilih file dari komputer Anda
3. Verifikasi bahwa semua file yang diperlukan telah diunggah.

> **Tip**
>
> Untuk hasil terbaik, sertakan diagram arsitektur, spesifikasi desain, dan dokumentasi teknis yang menggambarkan komponen dan aliran data yang relevan dengan keamanan sistem Anda.

Langkah 4: Memulai tinjauan desain

Setelah mengonfigurasi semua informasi yang diperlukan, mulailah analisis keamanan dokumen desain Anda.

1. Tinjau semua file dan pengaturan yang diunggah untuk memastikan akurasi.
2. Pilih Mulai tinjauan desain.
3. AWS Security Agent menganalisis dokumen desain Anda terhadap persyaratan keamanan yang diaktifkan.

> **Note**
>
> Proses peninjauan desain biasanya selesai dalam beberapa menit, tergantung pada jumlah dan ukuran file yang diunggah. Anda menerima temuan keamanan berdasarkan persyaratan keamanan organisasi Anda.

Langkah selanjutnya

Setelah memulai tinjauan desain Anda:

- Memantau kemajuan peninjauan di Aplikasi Web Agen
- Tinjau temuan keamanan
- Bagikan temuan dengan tim pengembangan Anda
- Alamat temuan keamanan yang teridentifikasi dalam desain Anda
- Perbarui dokumen desain dan kirim ulang jika diperlukan

Tinjau temuan dari tinjauan desain

Temuan tinjauan membantu Anda memahami persyaratan keamanan mana yang dipenuhi, mana yang perlu diperhatikan, dan tindakan apa yang harus diambil untuk meningkatkan postur keamanan desain Anda sebelum implementasi dimulai.

Dalam prosedur ini, Anda akan belajar cara mengakses, memfilter, dan menafsirkan temuan tinjauan desain untuk mengatasi temuan keamanan secara efektif.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Tinjauan desain yang lengkap
- Akses ke aplikasi web AWS Security Agent
- Keakraban dengan persyaratan keamanan yang diaktifkan organisasi Anda

Langkah 1: Akses ulasan desain

Arahkan ke tinjauan desain Anda untuk melihat temuan dan informasi ringkasan.

1. Masuk ke aplikasi web AWS Security Agent.
2. Arahkan ke bagian Ulasan desain.
3. Pilih ulasan desain yang ingin Anda periksa dari daftar.

> **Tip**
>
> Halaman detail tinjauan desain menampilkan ringkasan status tinjauan, tanggal penyelesaian, dan jumlah file yang ditinjau.

Langkah 2: Tinjau ringkasan temuan

Periksa ringkasan tingkat tinggi untuk memahami postur keamanan keseluruhan desain Anda.

1. Temukan bagian Ringkasan.
2. Tinjau jumlah untuk setiap kategori status kepatuhan: Sesuai, Non-compliant, Data tidak mencukupi, dan Tidak berlaku.

> **Note**
>
> Ringkasan memberikan hitungan untuk setiap jenis status, membantu Anda dengan cepat menilai jumlah temuan yang membutuhkan perhatian. Untuk penjelasan rinci tentang setiap status, lihat Langkah 4.

Langkah 3: Filter dan navigasikan temuan

Gunakan kemampuan pemfilteran dan pencarian untuk fokus pada temuan atau status kepatuhan tertentu.

1. Di bagian Tinjau temuan, cari kontrol filter.
2. Untuk memfilter berdasarkan status:
   a. Pilih menu tarik-turun status.
   b. Pilih status kepatuhan tertentu untuk hanya melihat temuan dengan status tersebut.
3. Untuk mencari persyaratan keamanan tertentu:
   a. Masukkan kata kunci di bidang pencarian.
   b. Hasil diperbarui secara otomatis saat Anda mengetik.
4. Gunakan kontrol pagination untuk menavigasi melalui beberapa halaman temuan.

> **Tip**
>
> Filter berdasarkan Non-compliantstatus terlebih dahulu untuk memprioritaskan temuan yang membutuhkan perhatian segera dalam desain Anda.

Langkah 4: Memahami status kepatuhan

Setiap temuan menampilkan status kepatuhan yang menunjukkan bagaimana desain Anda menangani persyaratan keamanan tertentu:

- Compliant — Desain Anda memenuhi persyaratan keamanan berdasarkan analisis
- Non-compliant— Desain Anda melanggar atau tidak memadai memenuhi persyaratan keamanan
- Data tidak mencukupi - File yang diunggah kekurangan informasi yang cukup untuk menentukan kepatuhan

- Tidak berlaku — Persyaratan keamanan tidak berlaku untuk desain sistem Anda

> **Important**
>
> Fokus pada Non-compliantdan Status data tidak mencukupi, karena ini memerlukan tindakan. Atasi temuan yang tidak sesuai dengan memperbarui desain Anda, dan selesaikan temuan data yang tidak mencukupi dengan mengunggah dokumentasi desain tambahan.

Langkah 5: Lihat detail temuan

Pilih temuan individu untuk melihat panduan pembenaran dan remediasi terperinci.

1. Dalam tabel temuan, pilih nama persyaratan keamanan.
2. Tinjau detail temuan, yang meliputi:
   - Persyaratan keamanan khusus yang sedang dievaluasi
   - Komentar yang menjelaskan mengapa temuan tersebut menerima status kepatuhannya, termasuk detail spesifik tentang apa yang hilang atau tidak sesuai
   - Panduan remediasi yang direkomendasikan untuk mengatasi temuan
   - Tautan ke dokumentasi atau standar internal organisasi Anda untuk persyaratan keamanan

> **Note**
>
> Komentar tersebut menjelaskan alasan AWS Security Agent dengan detail spesifik. Untuk temuan data yang tidak mencukupi, komentar mengidentifikasi informasi apa yang hilang, seperti “Dokumen desain tidak menyebutkan mekanisme otentikasi” atau “Tidak ada informasi yang ditemukan tentang enkripsi data saat istirahat.”

Langkah 6: Mengatasi temuan

Ambil tindakan atas temuan yang memerlukan perhatian untuk meningkatkan postur keamanan desain Anda.

Untuk Non-compliantemuan:

1. Tinjau panduan remediasi yang direkomendasikan.

2. Tinjau dokumentasi internal yang ditautkan untuk konteks tambahan.
3. Perbarui dokumen desain Anda untuk memenuhi persyaratan keamanan.
4. Dokumentasikan perubahan yang Anda buat untuk referensi future.

Untuk temuan data yang tidak mencukupi:

1. Baca komentar dengan seksama untuk memahami informasi spesifik apa yang hilang.
2. Buat atau perbarui dokumen desain dengan detail yang hilang.
3. Siapkan file yang diperbarui untuk pengiriman ulang.

Langkah selanjutnya

Setelah meninjau temuan desain Anda:

- Unduh laporan temuan sebagai file CSV untuk dibagikan dengan tim Anda
- Perbarui dokumen desain untuk mengatasi temuan yang tidak sesuai
- Buat dokumentasi tambahan untuk temuan data yang tidak mencukupi
- Bagikan temuan dengan tim pengembangan Anda untuk diskusi
- Kloning tinjauan desain ini untuk membuat tinjauan baru dengan dokumen asli yang dimuat sebelumnya, memungkinkan Anda memperbarui nama dan menjalankan analisis lagi untuk memverifikasi peningkatan
- Lanjutkan dengan implementasi untuk desain yang memenuhi persyaratan kepatuhan

Untuk informasi selengkapnya tentang mengelola persyaratan keamanan, lihatthe section called “Kelola persyaratan keamanan”.

Untuk informasi selengkapnya tentang membuat ulasan desain, lihatthe section called “Buat tinjauan desain”.

Buat model ancaman

Buat model ancaman di aplikasi web AWS Security Agent untuk menganalisis arsitektur aplikasi Anda dan mengidentifikasi ancaman keamanan. Model ancaman menghasilkan dua output utama: ikhtisar sistem yang menjelaskan bagaimana sistem Anda dibangun dan berperilaku, dan serangkaian

ancaman yang menggambarkan bagaimana sistem dapat diserang, masing-masing dengan tingkat keparahan, klasifikasi STRIDE, dan rekomendasi yang dapat ditindaklanjuti.

Model ancaman menerima dua jenis input, dan Anda dapat memberikan salah satu atau keduanya:

- Dokumen lingkup — Dokumen desain teknis, spesifikasi API, atau dokumentasi arsitektur yang menentukan fokus model ancaman. Ketika disediakan, agen mencakup analisisnya ke konteks yang dijelaskan dalam dokumen-dokumen ini. Anda dapat mengunggah file (DOC, DOCX, JPEG, MD, PDF, PNG, TXT), menyediakan URI S3, atau menghubungkan halaman Confluence melalui integrasi.
- Sumber - Kode sumber dari repositori yang terhubung (GitHub, Server GitHub Perusahaan,, GitLab GitLab Self-Managed, atau Bitbucket) atau dari lokasi S3. AWS Security Agent membaca kode sumber untuk memahami sistem yang ada. Saat digabungkan dengan dokumen lingkup, kode sumber memberikan konteks tentang implementasi saat ini.

Dalam prosedur ini, Anda membuat model ancaman dengan mengonfigurasi input dan izinnya, lalu memulai proses.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Akses ke aplikasi web AWS Security Agent
- Setidaknya salah satu dari berikut ini:
  - Sebuah repositori terhubung (GitHub, GitHub Enterprise Server,, GitLab GitLab Self-Managed, atau Bitbucket) untuk digunakan sebagai sumber (lihat) the section called “Aktifkan pemodelan ancaman”
  - Bucket S3 yang berisi kode sumber
  - Merancang dokumen untuk diunggah sebagai dokumen cakupan, atau integrasi Confluence

> **Tip**
>
> Jika Anda sudah memiliki repositori atau bucket S3 yang terhubung ke Ruang Agen (misalnya, melalui peninjauan kode atau pengaturan pengujian penetrasi), Anda dapat segera membuat model ancaman — tidak diperlukan pengaturan tambahan.

Bagaimana AWS Security Agent menjalankan model ancaman

Masukan yang Anda berikan menentukan cara AWS Security Agent menjalankan model ancaman. Ada tiga skenario.

Masukan yang Anda berikan	Bagaimana AWS Security Agent menjalankan model ancaman
Sumber saja (kode sumber)	AWS Security Agent menganalisis kode sumber untuk memahami struktur aplikasi Anda, mengklasifikasikan komponen, mengekstrak aliran data, membangun model ancaman, dan mengidentifikasi ancaman berdasarkan bagaimana sistem sebenarnya diterapkan.
Cakupan dokumen saja (dokumen desain)	AWS Security Agent menjalankan model ancaman yang berfokus pada desain yang dijelaskan dalam dokumen Anda. Ini mengident ifikasi ancaman berdasarkan arsitektur, aliran data, dan komponen yang dijelaskan dalam dokumen lingkup — berguna untuk pemodelan ancaman desain sebelum kode apa pun ada.
Baik dokumen sumber maupun ruang lingkup	AWS Security Agent mencakup model ancaman ke konteks yang dijelaskan dalam dokumen lingkup (misalnya, dokumen desain untuk fitur baru). Ini menggunakan kode sumber untuk memahami sistem Anda yang ada, kemudian mengancam model desain yang dijelaskan dalam dokumen lingkup — apakah desain itu sudah diterapkan atau belum.

Akses halaman model ancaman

Arahkan ke bagian pemodelan ancaman di aplikasi web.

1. Masuk ke aplikasi web AWS Security Agent.
2. Di bilah sisi kiri, pilih model Ancaman.
3. Anda melihat daftar model ancaman yang ada dengan judul, status run terbaru, dan jumlah ancaman berdasarkan tingkat keparahannya.

Buat model ancaman

Siapkan model ancaman baru dengan mengonfigurasi input dan izinnya.

1. Pada halaman Model ancaman, pilih Buat model ancaman.

Konfigurasikan detail model ancaman

Berikan judul untuk model ancaman Anda.

1. Di bidang Judul, masukkan nama deskriptif untuk model ancaman Anda.

> **Tip**
>
> Gunakan nama yang mengidentifikasi aplikasi atau layanan yang dimodelkan. Misalnya, “layanan penagihan” atau “checkout-api”.

Tambahkan dokumen lingkup

Berikan dokumen desain teknis yang menentukan fokus model ancaman, seperti dokumentasi arsitektur, spesifikasi API, atau proposal desain. Ketika dokumen lingkup disediakan, agen akan mencakup analisisnya ke konteks yang dijelaskan dalam dokumen ini. Dokumen cakupan bersifat opsional jika Anda menyediakan sumber.

1. Di bagian Dokumen Lingkup, tambahkan dokumen menggunakan satu atau beberapa metode berikut:
   - Unggah file — Unggah dokumen desain secara langsung (DOC, DOCX, JPEG, MD, PDF, PNG, atau TXT).
   - URI S3 — Masukkan URI S3 yang menunjuk ke dokumen yang disimpan dalam bucket S3 yang terhubung.
   - Halaman pertemuan — Jika Anda memiliki integrasi Confluence yang terhubung ke Ruang Agen Anda, pilih halaman dari ruang kerja Confluence Anda.

> **Tip**
>
> Untuk hasil terbaik, sertakan diagram arsitektur, spesifikasi API, dan dokumentasi teknis yang menjelaskan komponen sistem Anda, aliran data, dan batas kepercayaan.

Tambahkan sumber

Pilih kode sumber yang digunakan AWS Security Agent untuk memahami sistem yang ada. Ketika dikombinasikan dengan dokumen lingkup, kode sumber memberikan konteks tentang implementasi saat ini — agen menggunakannya untuk memahami apa yang sudah ada. Sumber bersifat opsional jika Anda menyediakan dokumen cakupan.

1. Di bagian Sumber, tambahkan kode sumber menggunakan satu atau beberapa metode berikut:

Repositori terintegrasi

Pilih dari repositori yang terhubung ke Ruang Agen Anda (GitHub, Server GitHub Perusahaan,, GitLab GitLab Self-Managed, atau Bitbucket).

1. Pilih kotak centang di samping setiap repositori yang ingin Anda sertakan sebagai sumber.
2. Gunakan bidang pencarian untuk menemukan repositori tertentu berdasarkan nama.

> **Note**
>
> Hanya repositori yang terhubung ke Ruang Agen Anda yang muncul di sini. Untuk menambahkan lebih banyak repositori, minta administrator Anda untuk memperbarui konfigurasi Ruang Agen.

Sumber S3

Tambahkan URI S3 yang menunjuk ke kode sumber yang disimpan dalam bucket S3 yang terhubung.

1. Masukkan URI S3 untuk setiap lokasi kode sumber yang ingin Anda sertakan.

Konfigurasikan sumber daya AWS

Pilih peran layanan IAM dan grup CloudWatch log opsional untuk model ancaman ini.

1. Di menu tarik-turun peran Layanan, pilih peran IAM dari peran layanan yang dikonfigurasi.

> **Note**
>
> Peran layanan harus memiliki izin untuk mengakses kode sumber Anda di S3 dan menulis ke CloudWatch log. Peran layanan dikonfigurasi selama penyiapan Ruang Agen di AWS Management Console.

2. (Opsional) Di dropdown grup Log, pilih grup CloudWatch log untuk menyimpan log eksekusi model ancaman.

Buat model ancaman

1. Tinjau konfigurasi Anda.
2. Pilih Buat model ancaman.

Anda diarahkan ke halaman detail model ancaman tempat Anda dapat memulai lari.

Jalankan model ancaman

Setelah membuat model ancaman, mulailah berlari untuk memulai analisis.

1. Pada halaman detail model ancaman, pilih Mulai jalankan.
2. AWS Security Agent mulai menganalisis sumber dan dokumen cakupan Anda.

Anda juga dapat memulai proses dari halaman daftar model Ancaman dengan memilih Mulai jalankan di sebelah model ancaman yang ingin Anda jalankan.

Pantau model ancaman yang dijalankan

Lacak kemajuan model ancaman Anda saat dijalankan.

Status eksekusi

Header halaman run menampilkan indikator status:

- Sedang berlangsung — Agen model ancaman sedang berjalan.
- Berhenti — Pembatalan diminta; agen sedang menyelesaikan tugasnya saat ini sebelum berhenti.

- Selesai — Lari selesai dengan sukses.
- Gagal - Jalankan mengalami kesalahan. Pesan kesalahan ditampilkan dengan detail. Mulai proses baru setelah menyelesaikan masalah.
- Berhenti - Jalankan dibatalkan oleh pengguna. Setiap ancaman yang dihasilkan sebelum pemberhentian dipertahankan.

Jalankan tab halaman

Pada halaman run detail, navigasikan antar tab:

- Ikhtisar — Lihat ringkasan run (ID pekerjaan, waktu mulai, status, durasi) dengan diagram lingkaran tingkat keparahan yang menunjukkan rincian ancaman berdasarkan tingkat keparahan (Kritis, Tinggi, Sedang, Rendah). Di bawah ringkasan, lihat diagram batang kategori ancaman STRIDE yang menunjukkan berapa banyak ancaman yang termasuk dalam setiap kategori STRIDE. Setelah proses selesai, lihat ikhtisar sistem yang dihasilkan oleh agen.
- Konfigurasi — Lihat sumber, dokumen, dan izin (peran layanan, grup CloudWatch log) yang digunakan untuk menjalankan ini.
- Preflight — Melihat status pemeriksaan preflight yang berjalan sebelum analisis ancaman dimulai, termasuk penyiapan infrastruktur pencatatan, validasi akses sumber S3 (bila berlaku), dan pengaturan lingkungan pengujian. Bilah kemajuan menunjukkan berapa banyak pemeriksaan yang telah diselesaikan.
- Log — Lihat daftar tugas yang dapat disaring yang dilakukan agen selama menjalankan. Pilih tugas apa pun untuk melihat keluaran log terperinci di panel samping.
- Ancaman - Lihat ancaman yang diidentifikasi selama dijalankan (lihatthe section called “Meninjau ancaman dari model ancaman”).

Jalankan sejarah

Setiap model ancaman mempertahankan sejarah semua proses. Pada halaman detail model ancaman:

- Tabel run mencantumkan semua proses dengan waktu mulai, status, durasi, jumlah ancaman, dan ID.
- Pilih tautan waktu mulai proses apa pun untuk melihat detail lengkapnya.

> **Tip**
>
> Re-run model ancaman yang sama setelah Anda memperbarui kode sumber atau merevisi dokumen cakupan Anda untuk melihat bagaimana ikhtisar sistem dan ancaman berubah dari waktu ke waktu.

Edit model ancaman

Untuk memodifikasi konfigurasi model ancaman Anda:

1. Pada halaman detail model ancaman, pilih Edit.
2. Perbarui judul, sumber, dokumen cakupan, atau sumber daya AWS sesuai kebutuhan.
3. Pilih Simpan perubahan.

> **Note**
>
> Mengedit model ancaman tidak memengaruhi proses yang telah diselesaikan sebelumnya. Mereka mempertahankan snapshot konfigurasi yang digunakan pada saat dijalankan.

Hapus model ancaman

Untuk menghapus model ancaman dan semua proses dan ancaman yang terkait:

1. Pada halaman detail model ancaman, buka menu tindakan dan pilih Hapus.
2. Konfirmasi penghapusan.

> **Important**
>
> Jika proses sedang berlangsung, Anda harus menghentikannya sebelum menghapus model ancaman.

Langkah selanjutnya

Setelah menjalankan model ancaman:

- Tinjau ikhtisar sistem dan ancaman (lihatthe section called “Meninjau ancaman dari model ancaman”)
- Re-run model ancaman setelah melakukan perubahan untuk memverifikasi ancaman ditangani
- Sesuaikan dokumen sumber dan cakupan Anda saat aplikasi Anda berkembang

Meninjau ancaman dari model ancaman

Setelah model ancaman berjalan selesai, tinjau ikhtisar sistem dan ancaman untuk memahami bagaimana aplikasi Anda dapat diserang dan apa yang harus dilakukan. Ikhtisar sistem adalah dokumen komprehensif yang menjelaskan arsitektur aplikasi Anda, batas kepercayaan, aliran data, dan postur keamanan. Setiap ancaman mencakup pernyataan, tingkat keparahan, klasifikasi STRIDE, aset yang terpengaruh, dan rekomendasi untuk mengatasinya.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Model ancaman yang lengkap dijalankan
- Akses ke aplikasi web AWS Security Agent

Langkah 1: Akses model ancaman yang dijalankan

Arahkan ke proses model ancaman Anda yang telah selesai.

1. Masuk ke aplikasi web AWS Security Agent.
2. Di bilah sisi kiri, pilih model Ancaman.
3. Pilih model ancaman yang ingin Anda periksa.
4. Di tabel run, pilih run selesai dengan memilih tautan waktu mulai.

Langkah 2: Tinjau ikhtisar sistem

Gambaran umum sistem adalah deskripsi komprehensif tentang cara AWS Security Agent memahami sistem Anda. Ini adalah dokumen terstruktur yang dapat mencakup:

- Tujuan — Apa yang dilakukan sistem dan siapa yang dilayaninya.
- Kemampuan — Fungsionalitas utama yang disediakan sistem.

- Maksud desain — Perubahan desain atau fitur yang dimodelkan dengan ancaman (saat dokumen cakupan disediakan).
- Arsitektur — Bagaimana sistem dibangun, termasuk pola penyebaran dan protokol komunikasi.
- Komponen — Tabel komponen sistem dengan tujuan dan interaksi kuncinya.
- Batas kepercayaan — Di mana konteks keamanan berubah, termasuk perlindungan apa yang ada di setiap persimpangan.
- Aliran data — Deskripsi terperinci tentang bagaimana data bergerak melalui sistem, termasuk protokol, kredensi, dan perlindungan di setiap langkah.
- Postur keamanan — Otentikasi saat ini, enkripsi, dan mekanisme kontrol akses.
- Aset sensitif — Data dan kredensil yang memerlukan perlindungan, dengan klasifikasi dan titik eksposurnya.
- Asumsi kunci — Security-relevant asumsi yang dibuat agen tentang sistem.

Untuk meninjau ikhtisar sistem:

1. Pilih tab Ikhtisar.
2. Tinjau bagian Run summary, yang menampilkan ID pekerjaan, waktu mulai, status, dan durasi.
3. Tinjau grafik tingkat Keparahan, yang menunjukkan rincian ancaman berdasarkan tingkat keparahan (Kritis, Tinggi, Sedang, Rendah) dengan jumlah dan persentase.
4. Tinjau bagan kategori Ancaman, yang menunjukkan berapa banyak ancaman yang termasuk dalam setiap kategori STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege).
5. Di bagian Ikhtisar sistem, baca analisis lengkap agen.

> **Tip**
>
> Jika ikhtisar sistem tidak mencerminkan sistem Anda secara akurat, perbaiki input Anda — tambahkan repositori yang relevan sebagai sumber atau unggah dokumen lingkup yang lebih lengkap — dan jalankan model ancaman lagi.

Langkah 3: Tinjau ancaman

Arahkan ke tab Ancaman untuk melihat semua ancaman yang diidentifikasi selama proses dijalankan.

1. Pilih tab Ancaman.
2. Ancaman ditampilkan sebagai daftar dengan setiap kartu yang menunjukkan judul ancaman, lencana tingkat keparahan, status, dan stempel waktu terakhir yang diperbarui. Anda dapat memfilter ancaman berdasarkan tingkat keparahan, status, atau pencarian berdasarkan judul.
3. Pilih ancaman dari daftar untuk melihat detail lengkapnya di panel kanan.

Tingkat keparahan ancaman

Setiap ancaman diberi tingkat keparahan:

- Kritis — Membutuhkan tindakan segera; eksploitasi dapat menyebabkan kompromi sistem penuh.
- Tinggi — Membutuhkan perhatian segera; eksploitasi dapat mengakibatkan dampak keamanan yang signifikan.
- Sedang — Harus ditangani dalam jangka waktu yang wajar; berkontribusi terhadap risiko keamanan secara keseluruhan.
- Rendah - Dapat diatasi sebagai bagian dari perawatan rutin; risiko langsung minimal.

Detail ancaman

Pilih ancaman untuk melihat detailnya di panel kanan. Rinciannya diatur ke dalam bagian-bagian berikut:

Baris metadata:

- Tingkat keparahan — Tingkat risiko yang ditetapkan oleh agen (Kritis, Tinggi, Sedang, atau Rendah).
- Kategori STRIDE — Klasifikasi ancaman: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, atau Elevation of Privilege.
- Dibuat di — Ketika ancaman diidentifikasi.
- Terakhir diperbarui - Saat ancaman terakhir diubah.

Bagian deskripsi:

- Pernyataan — Deskripsi bahasa alami tentang ancaman: apa yang dapat dilakukan sumber ancaman, apa dampaknya, dan kondisi apa yang memungkinkannya.
- Sumber — Aktor atau asal ancaman (misalnya, “pengguna yang diautentikasi” atau “penyerang eksternal”).
- Tindakan — Apa yang dapat dilakukan sumber ancaman (misalnya, “menyuntikkan kueri SQL ke dalam parameter pencarian”).
- Dampak — Konsekuensi langsung dari tindakan ancaman (misalnya, “akses tidak sah ke catatan pelanggan”).

Bagian referensi:

- Aset yang terpengaruh - Aset spesifik yang dipengaruhi oleh ancaman (misalnya, “catatan pembayaran pelanggan” atau “tabel DynamoDB”).
- Prasyarat — Kondisi yang harus benar agar ancaman dapat dieksploitasi.
- Sasaran yang terkena dampak — Sasaran keamanan yang terpengaruh: Kerahasiaan, Integritas, Ketersediaan, Otorisasi, Otentikasi, atau. Non-repudiation
- Bukti — Jalur file kode sumber yang mendukung ancaman, menautkan kembali ke file tertentu di repositori Anda.
- Anchor — Referensi kode yang menghubungkan ancaman ke node tertentu dalam kode sumber Anda.

Bagian rekomendasi:

- Rekomendasi — Panduan yang dapat ditindaklanjuti untuk mengatasi ancaman.

Langkah 4: Buat ancaman secara manual

Anda dapat menambahkan ancaman yang tidak diidentifikasi agen — misalnya, ancaman yang ditemukan selama peninjauan manual atau dari sumber eksternal.

1. Pada tab Ancaman dari proses yang telah selesai, pilih Buat ancaman.
2. Isi detail ancaman:
   - Pernyataan — Deskripsi bahasa alami tentang ancaman.
   - Keparahan — Pilih Kritis, Tinggi, Sedang, atau Rendah.
   - Sumber — Aktor atau asal ancaman.

- Prasyarat — Kondisi yang diperlukan agar ancaman dapat dieksploitasi.
- Tindakan — Apa yang dapat dilakukan sumber ancaman.
- Dampak — Konsekuensi langsung dari tindakan ancaman.
- Aset yang terpengaruh - Aset spesifik yang terpengaruh (dipisahkan koma).
- Tujuan keamanan yang terkena dampak — Pilih dari Kerahasiaan, Integritas, Ketersediaan, Otorisasi, Otentikasi, atau. Non-repudiation
- Kategori STRIDE - Pilih kategori yang berlaku.
- Rekomendasi — Panduan untuk mengatasi ancaman.

3. Pilih Buat.

Ancaman yang dibuat secara manual muncul dalam daftar ancaman bersama ancaman yang dihasilkan agen.

Langkah 5: Edit dan triase ancaman

Saat meninjau ancaman, Anda dapat mengedit detailnya dan memperbarui statusnya untuk melacak kemajuan.

1. Pilih ancaman dari daftar.
2. Pilih ikon edit di panel detail ancaman untuk membuka formulir edit.
3. Anda dapat memodifikasi bidang berikut:
   - Status - Lacak siklus hidup ancaman:
     - Terbuka — Ancaman diakui dan perlu diperhatikan (default).
     - Terselesaikan - Anda telah memperbaiki masalah.
     - Diberhentikan — Anda meninjau ancaman dan memutuskan bahwa itu tidak berlaku.
   - Keparahan — Sesuaikan tingkat keparahan jika penilaian agen tidak sesuai dengan konteks Anda.
   - Pernyataan — Perbaiki deskripsi ancaman.
   - Sumber, Prasyarat, Tindakan, Dampak — Perbarui detail ancaman berdasarkan pengetahuan domain Anda.
   - Aset yang terpengaruh - Menambah atau menghapus aset yang terpengaruh (dipisahkan koma).
   - Tujuan keamanan yang terpengaruh - Pilih tujuan keamanan yang terpengaruh (Kerahasiaan, Integritas, Ketersediaan, Otorisasi, Otentikasi,). Non-repudiation

4. Pilih Simpan untuk menerapkan perubahan Anda.

Langkah 6: Unduh laporan

Setelah proses selesai, Anda dapat mengunduh laporan PDF yang merangkum ikhtisar sistem dan semua ancaman yang diidentifikasi.

1. Pada halaman run yang sudah selesai, pilih Hasilkan laporan.
2. PDF diunduh ke komputer Anda.

Langkah 7: Tinjau cek dan log preflight

Jika Anda perlu menyelidiki bagaimana agen mencapai kesimpulan atau men-debug kegagalan sebagian:

1. Pilih tab Preflight untuk melihat status pemeriksaan preflight yang berjalan sebelum analisis ancaman dimulai. Setiap pemeriksaan menunjukkan statusnya (lengkap, sedang berlangsung, atau tertunda), dan bilah kemajuan menunjukkan berapa banyak pemeriksaan yang telah diselesaikan. Anda dapat memperluas pemeriksaan lengkap untuk melihat keluaran log detailnya.
2. Pilih tab Log untuk melihat daftar tugas yang dapat difilter yang dilakukan agen selama menjalankan. Pilih tugas apa pun dari daftar untuk melihat keluaran log terperinci di panel samping.

Langkah selanjutnya

Setelah meninjau hasil model ancaman Anda:

- Atasi ancaman tingkat keparahan tinggi terlebih dahulu berdasarkan rekomendasi agen
- Perbarui status ancaman saat Anda menerapkan perbaikan
- Jalankan model ancaman baru untuk memverifikasi perubahan Anda mengatasi ancaman yang diidentifikasi
- Sesuaikan dokumen sumber dan cakupan Anda saat aplikasi Anda berkembang (lihat) the section called “Buat model ancaman”

Tinjau temuan keamanan kode dalam permintaan tarik

Setelah mengaktifkan peninjauan kode untuk permintaan tarik untuk repositori Anda, AWS Security Agent secara otomatis menganalisis permintaan tarik dan memposting temuan keamanan langsung di penyedia kontrol sumber Anda. Hal ini memungkinkan pengembang untuk mengatasi masalah keamanan dalam alur kerja normal mereka tanpa meninggalkan permintaan tarik.

> **Note**
>
> Halaman ini berlaku untuk permintaan GitHub tarik, permintaan GitLab gabungan, dan permintaan tarik Bitbucket. Pengalamannya serupa di semua penyedia.

Cara kerja peninjauan kode dalam permintaan tarik

Saat Anda mengirimkan permintaan tarik (atau menggabungkan permintaan GitLab) di repositori dengan peninjauan kode diaktifkan, AWS Security Agent secara otomatis memulai analisis.

1. Pemicu analisis permintaan tarik - Tinjauan kode dipicu ketika permintaan tarik ditandai sebagai “Siap untuk ditinjau” di repositori tempat Anda mengaktifkan kemampuan peninjauan kode. Permintaan tarik draf tidak dianalisis.
2. Pengakuan analisis - Ketika AWS Security Agent mulai menganalisis permintaan tarik Anda, ia memposting komentar awal: “AWS Security Agent sedang menganalisis kode Anda...” Ini memberi tahu Anda bahwa analisis telah dimulai dan sedang berlangsung.
3. Penyelesaian tinjauan - Setelah analisis selesai, AWS Security Agent memposting ulasan ke permintaan tarik Anda dengan hasilnya. Semua temuan keamanan dikumpulkan bersama dalam satu tinjauan untuk menjaga permintaan tarik Anda tetap teratur dan meminimalkan pemberitahuan.

Memahami hasil tinjauan kode

AWS Security Agent menyediakan berbagai jenis hasil tergantung pada apa yang ditemukan selama analisis.

Ketika masalah keamanan ditemukan

Jika AWS Security Agent mengidentifikasi masalah keamanan dalam perubahan kode Anda, AWS akan memposting ulasan yang mencakup:

- Ringkasan - Tinjauan tingkat tinggi dari semua temuan keamanan, menjelaskan jenis masalah yang diidentifikasi dan potensi dampaknya
- Temuan individu - Temuan keamanan terperinci muncul sebagai komentar berulir di bawah tinjauan utama, dengan setiap temuan termasuk:
  - Deskripsi masalah keamanan
  - Lokasi dalam kode Anda tempat masalah ditemukan
  - Panduan remediasi yang menjelaskan cara mengatasi masalah ini
  - Konteks yang relevan berdasarkan pengaturan peninjauan kode Anda (pelanggaran persyaratan keamanan, kerentanan umum, atau keduanya)

> **Note**
>
> Jenis masalah keamanan yang dianalisis bergantung pada pengaturan peninjauan kode Anda. Jika Anda mengonfigurasi validasi persyaratan keamanan, temuan akan merujuk persyaratan keamanan khusus organisasi Anda. Jika Anda mengonfigurasi temuan kerentanan keamanan, temuan akan mengidentifikasi kerentanan keamanan umum. Untuk informasi selengkapnya tentang pengaturan tinjauan kode, lihatthe section called “Aktifkan tinjauan kode permintaan tarik untuk GitHub repositori”.

Ketika tidak ada masalah keamanan yang ditemukan

Jika AWS Security Agent menyelesaikan analisis dan tidak menemukan masalah keamanan dalam perubahan kode Anda, AWS akan memposting komentar: “Tidak ada masalah yang diidentifikasi.” Ini mengonfirmasi bahwa peninjauan berhasil diselesaikan dan perubahan kode Anda tidak memicu temuan keamanan apa pun berdasarkan pengaturan peninjauan kode yang dikonfigurasi.

Menanggapi temuan keamanan

Setelah meninjau temuan keamanan yang diposting oleh AWS Security Agent, Anda dapat mengambil tindakan langsung di penyedia kontrol sumber Anda.

- Temuan alamat - Perbarui kode Anda berdasarkan panduan remediasi yang diberikan dalam temuan, lalu dorong komit baru ke permintaan tarik. AWS Security Agent akan menganalisis kode yang diperbarui.

- Selesaikan percakapan - Setelah menangani temuan keamanan, tandai percakapan sebagai diselesaikan untuk melacak kemajuan Anda.

> **Tip**
>
> Setiap temuan mencakup panduan remediasi khusus yang disesuaikan dengan masalah keamanan yang diidentifikasi. Tinjau panduan ini dengan seksama untuk memahami risiko keamanan dan cara mengatasinya secara efektif.

Memfilter temuan tinjauan kode

Anda dapat menyesuaikan cara AWS Security Agent menganalisis kode Anda dengan menambahkan `filtering.md` file ke repositori Anda. File ini memungkinkan Anda untuk mengurangi positif palsu dengan memberikan konteks tentang basis kode Anda dan mengecualikan file atau folder dari analisis.

Membuat file penyaringan

Buat file bernama `filtering.md` di `.awssecurityagent` direktori di root repositori Anda:

```
.awssecurityagent/filtering.md
```

AWS Security Agent membaca file ini dari cabang utama repositori Anda (misalnya, `main` atau`mainline`) saat menganalisis permintaan tarik.

Struktur file

`filtering.md`File menggunakan format Markdown standar dengan bagian tertentu yang dikenali AWS Security Agent. File harus menyertakan `Code Review` judul diikuti oleh salah satu atau kedua bagian berikut: `IgnorePatterns` dan `ContextHints` (tidak ada spasi).

Contoh berikut menunjukkan struktur lengkap `filtering.md` file:

```
# 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.
```

Abaikan pola

`IgnorePatterns`Bagian ini menentukan file dan folder yang harus dilewati AWS Security Agent selama peninjauan kode. Gunakan `glob patterns` untuk menentukan jalur mana yang akan dikecualikan dari analisis.

Persyaratan format:

- Setiap pola harus pada garisnya sendiri.
- Pisahkan setiap pola dengan garis kosong di antara mereka. Ini memastikan file dirender dengan benar saat dilihat di GitHub atau alat peninjauan kode.
- Pola mengikuti format glob standar. Misalnya, `**/*.md` mencocokkan semua file penurunan harga, dan `/myapp/src/**/*.snap` mencocokkan semua `.snap` file di dalam `/myapp/src/` folder di root.
- Kami mendukung hingga 1000 pola abaikan di bagian ini.

Petunjuk konteks

`ContextHints`Bagian ini menyediakan konteks tambahan tentang basis kode Anda yang membantu AWS Security Agent membuat penilaian yang lebih akurat. Gunakan petunjuk konteks untuk menjelaskan keputusan arsitektur, pengecualian keamanan, atau informasi lain yang mungkin memengaruhi bagaimana temuan ditafsirkan.

Persyaratan format:

- Setiap petunjuk harus dimulai dengan tanda hubung (`-`) diikuti oleh spasi.

- Tulis setiap petunjuk sebagai satu baris teks bentuk bebas yang dibatasi hingga 500 karakter.
- Setiap petunjuk harus menjelaskan satu bagian tertentu dari konteks tentang basis kode Anda.
- Kami mendukung hingga 20 petunjuk konteks di bagian ini.

Petunjuk konteks diterapkan setelah AWS Security Agent menyelesaikan analisis awalnya, membantu memfilter temuan yang tidak berlaku untuk kasus penggunaan spesifik Anda.

Langkah selanjutnya

Setelah meninjau temuan keamanan kode:

- Perbarui kode Anda berdasarkan panduan remediasi
- Dorong komitmen baru untuk memicu analisis ulang perubahan Anda
- Sesuaikan pengaturan peninjauan kode jika diperlukan (lihatthe section called “Aktifkan tinjauan kode permintaan tarik untuk GitHub repositori”)
- Tinjau persyaratan keamanan organisasi Anda untuk memahami kriteria validasi
- Pertimbangkan pengujian penetrasi untuk validasi keamanan komprehensif dari aplikasi yang digunakan

Buat ulasan kode

Buat ulasan kode di aplikasi web AWS Security Agent untuk memindai repositori kode sumber dan sumber S3 Anda untuk mencari kerentanan keamanan. Tinjauan kode melakukan analisis statis komprehensif di seluruh basis kode Anda, mengidentifikasi masalah keamanan, dan memberikan panduan remediasi.

Tidak seperti tinjauan kode berbasis permintaan tarik yang menganalisis perubahan kode individual (lihatthe section called “Tinjau temuan keamanan kode dalam permintaan tarik”), tinjauan kode sesuai permintaan memindai kode sumber lengkap Anda untuk mengidentifikasi kerentanan keamanan dan memvalidasi kepatuhan terhadap persyaratan keamanan organisasi Anda.

Dalam prosedur ini, Anda akan membuat tinjauan kode dengan memilih input kode sumber, mengonfigurasi izin, dan menjalankan tinjauan.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Akses ke aplikasi web AWS Security Agent
- Setidaknya satu GitHub repositori terhubung atau bucket S3 di Ruang Agen Anda

> **Tip**
>
> Jika Anda sudah memiliki GitHub repositori yang terhubung ke Ruang Agen Anda, peninjauan kode siap digunakan — tidak diperlukan pengaturan tambahan. Pilih Mulai di aplikasi web dari kartu peninjauan Kode di halaman Ruang Agen Anda, atau luncurkan aplikasi web secara langsung.
> Jika Anda perlu menghubungkan sumber tambahan atau mengkonfigurasi ember S3, lihat. the section called “Aktifkan peninjauan kode”

Akses halaman ulasan kode

Arahkan ke bagian ulasan kode di aplikasi web.

1. Masuk ke aplikasi web AWS Security Agent.
2. Di bilah sisi kiri, pilih Ulasan kode.
3. Anda melihat daftar ulasan kode yang ada dengan informasi sumbernya, status run terakhir, dan ringkasan temuan.

Buat ulasan kode

Siapkan tinjauan kode baru dengan mengonfigurasi input dan izin kode sumbernya.

1. Pada halaman Ulasan kode, pilih Buat ulasan kode.

Konfigurasikan detail ulasan kode

Berikan judul dan pilih kode sumber untuk ditinjau.

1. Di bidang Judul, masukkan nama deskriptif untuk tinjauan kode Anda.

> **Tip**
>
> Gunakan nama yang mengidentifikasi aplikasi, repositori, atau ruang lingkup tinjauan. Misalnya, “billing-service-security-review” atau “infrastruktur-kode-audit”.

2. Di bagian Sumber, pilih input kode sumber untuk ulasan ini. Pilih dari dua tab:

GitHub repositori

Pilih dari repositori yang terhubung ke Ruang Agen Anda.

1. Pilih tab GitHub repositori.
2. Dalam tabel Repositori terintegrasi, pilih kotak centang di samping setiap repositori yang ingin Anda sertakan dalam ulasan.
3. Gunakan bidang pencarian untuk menemukan repositori tertentu berdasarkan nama.

> **Note**
>
> Hanya repositori yang terhubung ke Ruang Agen Anda melalui konfigurasi peninjauan kode yang muncul di sini. Untuk menambahkan lebih banyak repositori, pilih Kelola di konsol Admin Anda atau minta administrator Anda untuk memperbarui konfigurasi Ruang Agen.

Sumber S3

Pilih file ZIP dari bucket S3 yang terhubung ke Ruang Agen Anda. Administrator Ruang Agen Anda mengonfigurasi bucket S3 mana yang tersedia. File ZIP apa pun yang disimpan di salah satu bucket tersebut dapat digunakan sebagai sumber untuk tinjauan kode.

1. Pilih tab sumber S3.
2. Masukkan URI S3 dari setiap file ZIP yang ingin Anda sertakan dalam ulasan. Anda dapat menambahkan hingga 30 sumber S3.

> **Note**
>
> Sumber S3 harus berupa file ZIP yang disimpan dalam bucket S3 yang terhubung ke Ruang Agen Anda. Untuk membuat ember tambahan tersedia, lihatthe section called “Aktifkan peninjauan kode”.

Mengatur Konfigurasi Izin

Pilih peran layanan IAM dan grup CloudWatch log opsional untuk tinjauan kode ini.

1. Di bagian Izin, cari dropdown peran Layanan.
2. Pilih peran IAM dari peran layanan yang dikonfigurasi.

> **Note**
>
> Peran layanan harus memiliki izin untuk mengakses kode sumber Anda di S3 dan menulis ke CloudWatch log, dan sumber daya AWS lainnya yang diperlukan untuk peninjauan kode. Peran layanan dikonfigurasi selama penyiapan peninjauan kode di AWS Management Console.

3. (Opsional) Di dropdown grup CloudWatch log, pilih grup log untuk menyimpan log eksekusi peninjauan kode.

> **Note**
>
> Jika Anda tidak memilih grup log, AWS Security Agent akan membuat grup log default untuk menyimpan log tinjauan kode.

Konfigurasikan remediasi kode otomatis

Aktifkan remediasi otomatis agar AWS Security Agent menghasilkan perbaikan kode untuk semua temuan segera setelah peninjauan selesai.

1. Di bagian Remediasi kode otomatis, pilih kotak centang Aktifkan perbaikan kode otomatis.

Cara AWS Security Agent memberikan perbaikan tergantung pada sumbernya:

- GitHub Repositori pribadi — AWS Security Agent mengirimkan permintaan tarik dengan perbaikan ke repositori.
- GitHub Repositori publik — Untuk menghindari pengungkapan kerentanan sebelum diperbaiki, AWS Security Agent tidak membuka permintaan tarik. Sebaliknya, itu melampirkan perbedaan yang disarankan ke temuan yang dapat Anda unduh dari aplikasi web dan mendaftar secara pribadi.
- Sumber S3 — Remediasi kode tidak tersedia. Tinjau detail temuan dan terapkan perbaikan secara manual.

Important

Permintaan tarik remediasi yang dikirimkan ke repositori pribadi dapat dilihat oleh semua orang dengan akses baca. Tinjau perubahan sebelum menggabungkan. Remediasi kode otomatis hanya tersedia ketika GitHub repositori dipilih sebagai sumber.

Note

Saat dinonaktifkan, Anda masih dapat memicu remediasi kode secara manual untuk GitHub-sourced temuan individual setelah peninjauan selesai.

Konfigurasikan validasi simulasi

Aktifkan validasi simulasi untuk mengonfirmasi secara dinamis apakah kerentanan yang ditemukan dapat dieksploitasi dalam aplikasi yang sedang berjalan.

1. Di bagian Validasi simulasi, pilih kotak centang Aktifkan validasi simulasi.

Saat diaktifkan, AWS Security Agent menyediakan lingkungan simulasi setelah analisis statis selesai. Ini onboard kode sumber Anda, memulai layanan di dalam lingkungan, dan mencoba untuk mengeksploitasi kerentanan yang ditemukan selama pemindaian. Temuan kemudian diperbarui dengan status validasi yang menunjukkan apakah kerentanan berhasil dieksploitasi.

> **Note**
>
> Validasi simulasi saat ini hanya tersedia untuk aplikasi dockerizable mandiri. Ketika beberapa repositori dipilih sebagai sumber, validasi simulasi tidak tersedia.

> **Important**
>
> Validasi simulasi menambahkan waktu pemrosesan ke proses peninjauan kode Anda. Langkah validasi menyediakan lingkungan dan menjalankan upaya eksploitasi. Ini biasanya memakan waktu 1-3 jam tergantung pada jumlah temuan dan kompleksitas aplikasi.

Tujuan jaringan yang diizinkan

Lingkungan simulasi memiliki akses jaringan keluar yang dibatasi ke daftar izin yang telah ditentukan sebelumnya. Aplikasi Anda dapat mencapai domain berikut selama validasi:

Tabel berikut mencantumkan domain yang diizinkan dan tujuannya.

Domain	Tujuan
`.amazonaws.com` , `.aws.amazon.com`	Layanan AWS
`.public.ecr.aws`	Amazon ECR Public
`.docker.com` , `.docker.io`	Hub Docker
`.github.com` , `.githubusercontent.com`	GitHub
`.gitlab.com`	GitLab
`.npmjs.com` , `.npmjs.org`	registri npm

Domain	Tujuan
`.pypi.org` , `.pypi.python.org` , `.pythonhosted.org`	Indeks Paket Python
`.crates.io` , `.rustup.rs`	Paket karat
`.maven.org` , `.gradle.org`	Java/Gradle paket
`.nuget.org`	.NET paket
`.rubygems.org` , `.ruby-lang.org`	Paket Ruby
`.golang.org` , `.pkg.go.dev` , `.goproxy.io`	Paket Go
`.nodejs.org` , `.yarnpkg.com`	Node.js
`.alpinelinux.org` , `.debian.org` , `.ubuntu.com` , `.centos.org` , `.fedoraproject.org`	Repositori distribusi Linux
`.cloudfront.net`	CloudFront distribusi
`.google.com` , `.googleapis.com`	API Google
`.microsoft.com` , `.visualstudio.com`	Layanan Microsoft
`.sourceforge.net` , `.bitbucket.org`	Hosting sumber

> **Note**
>
> Jika aplikasi Anda memerlukan akses jaringan ke domain yang tidak ada dalam daftar ini, koneksi akan diblokir oleh firewall jaringan. Aplikasi yang bergantung pada API atau layanan eksternal yang tidak tercantum di atas mungkin tidak dimulai dengan benar di lingkungan simulasi.

Buat ulasan kode

1. Tinjau konfigurasi Anda untuk memastikan akurasi.
2. Pilih Buat ulasan kode.

Anda diarahkan ke halaman detail peninjauan kode tempat Anda dapat memulai peninjauan.

Jalankan tinjauan kode

Setelah membuat tinjauan kode, mulailah menjalankan untuk memulai analisis.

1. Pada halaman detail peninjauan kode, pilih Mulai peninjauan.
2. AWS Security Agent mulai menganalisis kode sumber Anda.

Anda juga dapat memulai tinjauan dari halaman daftar ulasan Kode dengan memilih Mulai tinjauan di samping tinjauan kode yang ingin Anda jalankan.

Memantau peninjauan kode yang dijalankan

Lacak kemajuan tinjauan kode Anda saat dijalankan.

Tinjau fase lari

Sebuah tinjauan kode berjalan melalui fase-fase berikut, ditampilkan sebagai indikator kemajuan:

1. Preflight — AWS Security Agent memvalidasi akses ke kode sumber Anda dan menyiapkan lingkungan pengujian. Pemeriksaan preflight meliputi:
   - Penyiapan infrastruktur layanan
   - Validasi akses sumber S3

- Pengaturan lingkungan pengujian

2. Analisis statis — AWS Security Agent memindai kode sumber Anda untuk mengetahui kerentanan keamanan dan pelanggaran persyaratan.
3. Validasi simulasi (opsional) — Jika validasi simulasi diaktifkan, AWS Security Agent menyediakan lingkungan simulasi dan menerapkan aplikasi Anda. Kemudian mencoba untuk mengeksploitasi kerentanan yang ditemukan selama analisis statis. Langkah ini menegaskan apakah temuan dapat dieksploitasi dalam runtime target. Fase ini hanya muncul ketika Anda mengaktifkan validasi simulasi selama pembuatan tinjauan kode.
4. Finalisasi — AWS Security Agent mengkompilasi temuan dan menghasilkan ringkasan hasil.

Lihat detail run

Pada halaman run detail, navigasikan antar tab untuk memantau kemajuan:

- Code review run — Lihat ringkasan run termasuk run ID, waktu pembuatan, status, durasi, jam tugas, rincian tingkat keparahan, dan bagan tipe risiko.
- Preflight — Lihat progres pemeriksaan preflight dan status setiap langkah validasi.
- Log peninjauan kode — Melihat tugas AWS Security Agent yang diidentifikasi dan dilakukan selama peninjauan, dengan log tugas terperinci untuk setiap langkah.
- Validasi simulasi — Saat validasi simulasi diaktifkan, lihat status penyediaan, tugas validasi untuk temuan individu, dan hasilnya.
- Temuan — Lihat temuan keamanan setelah peninjauan selesai (lihatthe section called “Tinjau temuan dari tinjauan kode”).

Jalankan sejarah

Setiap tinjauan kode menyimpan riwayat semua proses. Pada halaman detail ulasan kode:

- Bagian Lari terbaru menunjukkan proses terbaru dengan waktu mulai, status, durasi, dan ID.
- Tabel Semua berjalan mencantumkan semua proses sebelumnya dengan waktu mulai, status, durasi, ringkasan temuan, dan ID mereka.
- Pilih Monitor run untuk melihat detail proses aktif terbaru.
- Pilih tautan waktu mulai proses apa pun untuk melihat detail lengkapnya.

Langkah selanjutnya

Setelah menjalankan tinjauan kode:

- Tinjau temuan keamanan dan panduan remediasinya (lihatthe section called “Tinjau temuan dari tinjauan kode”)
- Memulihkan temuan melalui permintaan tarik otomatis atau perbaikan manual (lihat) the section called “Remediasi temuan tinjauan kode”
- Jalankan ulasan tambahan setelah menerapkan perbaikan untuk memverifikasi remediasi
- Sesuaikan konfigurasi atau sumber peninjauan kode Anda saat basis kode Anda berkembang

Tinjau temuan dari tinjauan kode

Setelah peninjauan kode selesai, tinjau ringkasan proses dan temuan keamanan untuk memahami kerentanan dalam kode sumber Anda. Setiap temuan berisi deskripsi, peringkat keparahan, lokasi kode, penalaran risiko, dan perbaikan yang disarankan untuk membantu Anda memprioritaskan dan mengatasi masalah keamanan.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Jalankan peninjauan kode yang sudah selesai atau sedang berlangsung
- Akses ke aplikasi web AWS Security Agent

Langkah 1: Akses peninjauan kode yang dijalankan

Arahkan ke proses peninjauan kode lengkap Anda untuk melihat ringkasan dan temuan.

1. Masuk ke aplikasi web AWS Security Agent.
2. Di bilah sisi kiri, pilih Ulasan kode.
3. Pilih ulasan kode yang ingin Anda periksa.
4. Dalam tabel Semua berjalan, pilih run selesai dengan mengklik tautan waktu mulai. Atau, pilih Monitor run untuk melihat proses terbaru.

Langkah 2: Pantau kemajuan lari

Lacak kemajuan peninjauan kode Anda menggunakan indikator langkah.

1. Temukan indikator langkah horizontal di bawah header halaman.
2. Tinjau status setiap fase:
   - Preflight — Memvalidasi akses ke kode sumber Anda dan mengatur lingkungan pemindaian. Pemeriksaan mencakup penyiapan infrastruktur layanan, validasi akses sumber S3, dan pengaturan lingkungan pengujian, yang mencakup pemeriksaan GitHub akses.
   - Analisis statis — Memindai kode sumber Anda untuk kerentanan keamanan dan pelanggaran persyaratan.
   - Validasi simulasi (opsional) - Ketika diaktifkan, menyediakan lingkungan simulasi dan upaya untuk mengeksploitasi kerentanan yang ditemukan terhadap aplikasi yang sedang berjalan.
   - Finalisasi — Menyusun temuan dan menghasilkan ringkasan hasil.

> **Note**
>
> Setiap langkah menampilkan indikator status (Selesai atau Sedang berlangsung). Lari selesai ketika semua fase menunjukkan Selesai. Langkah validasi simulasi hanya muncul saat Anda mengaktifkan validasi simulasi selama pembuatan tinjauan kode.

Langkah 3: Tinjau ringkasan run

Arahkan ke tab Jalankan peninjauan kode untuk melihat hasil tingkat tinggi.

1. Pilih tab Jalankan tinjauan kode.
2. Bagian Run summary menyediakan status run, durasi, dan detail tingkat tinggi lainnya. Ini juga menyediakan dasbor temuan keamanan yang dikategorikan berdasarkan tingkat keparahan dan jenis risiko.
3. Gambaran umum aplikasi oleh AWS Security Agent memberikan ringkasan pemindaian tinjauan kode saat proses selesai

Langkah 4: Tinjau hasil preflight

Arahkan ke tab Preflight untuk memverifikasi bahwa semua pemeriksaan akses sumber telah lolos.

1. Pilih tab Preflight.
2. Tinjau indikator kemajuan Preflight yang menunjukkan jumlah pemeriksaan yang diselesaikan.
3. Verifikasi bahwa setiap pemeriksaan menunjukkan status keberhasilan:
   - Penyiapan infrastruktur layanan - Mengonfirmasi lingkungan pengujian sudah siap
   - Validasi Akses Sumber S3 - Mengonfirmasi akses ke kode sumber S3 (jika ada)
   - Pengaturan Testing Environment - Mengonfirmasi lingkungan analisis dikonfigurasi

> **Note**
>
> Jika pemeriksaan preflight gagal, proses tidak akan melanjutkan ke analisis statis. Tinjau detail pemeriksaan untuk mengidentifikasi dan menyelesaikan masalah akses, lalu mulai proses baru.

Langkah 5: Tinjau log ulasan kode

Arahkan ke tab log peninjauan Kode untuk melihat tugas AWS Security Agent yang dilakukan selama analisis.

1. Pilih tab Log ulasan kode.
2. Jelajahi daftar tugas di panel kiri.
3. Pilih tugas untuk melihat log detailnya di panel kanan.

> **Tip**
>
> Gunakan bidang pencarian untuk memfilter tugas berdasarkan nama. Log memberikan wawasan tentang apa yang diperiksa AWS Security Agent dan bagaimana hal itu mencapai kesimpulannya.

Langkah 6: Arahkan ke temuan

Pilih tab Temuan untuk melihat semua temuan keamanan dari proses.

> **Note**
>
> Filter kepercayaan default
>
> Secara default, Anda hanya melihat temuan dengan kepercayaan agen yang tinggi. Untuk juga menunjukkan temuan dengan kepercayaan agen Sedang atau Rendah dan positif palsu, matikan sakelar Menyembunyikan temuan yang tidak diverifikasi.

1. Pilih tab Temuan.
2. Temuan ditampilkan dalam tampilan terpisah dengan daftar temuan di sebelah kiri dan detail temuan yang dipilih di sebelah kanan.
3. Tinjau informasi yang ditampilkan pada setiap kartu pencari:
   - Menemukan judul — Nama deskriptif yang merangkum masalah keamanan
   - Lencana keparahan — indikator Color-coded keparahan:
     - Kritis (merah) — Membutuhkan tindakan segera; eksploitasi dapat menyebabkan kompromi sistem
     - Tinggi (merah) - Membutuhkan perhatian segera; eksploitasi dapat mengakibatkan dampak keamanan yang signifikan
     - Sedang (oranye) - Harus ditangani dalam jangka waktu yang wajar; berkontribusi terhadap risiko keamanan secara keseluruhan
     - Rendah (kuning) - Dapat ditangani sebagai bagian dari perawatan rutin; risiko langsung minimal
   - Terakhir diperbarui - Stempel waktu kapan temuan terakhir diperbarui

Langkah 7: Tinjau detail temuan

Pilih temuan individu untuk melihat informasi komprehensif tentang setiap kerentanan.

1. Pilih temuan di panel kiri untuk menampilkan detailnya di panel kanan.
2. Tinjau tindakan yang tersedia:
   - Selesaikan temuan - Tandai temuan sebagai diselesaikan setelah Anda mengatasinya
   - Remediate code — Hasilkan permintaan tarik dengan perbaikan (tersedia untuk GitHub sumber)

3. Berikan umpan balik menggunakan Apakah temuan ini akurat? — Pilih Ya atau Tidak untuk membantu meningkatkan akurasi analisis masa depan.
4. Tinjau atribut kunci di bagian Ikhtisar:
   - Keyakinan agen — Tingkat kepercayaan AWS Security Agent dalam temuan ini
   - Keparahan — Tingkat risiko dengan lencana kode warna
   - Jenis Risiko — Kategori risiko keamanan
   - Status validasi - Saat validasi simulasi diaktifkan, menunjukkan apakah temuan tersebut dikonfirmasi dapat dieksploitasi di lingkungan berjalan:
     - Dikonfirmasi - Kerentanan berhasil dieksploitasi di lingkungan simulasi
     - Tidak direproduksi - Kerentanan tidak dapat dieksploitasi dalam runtime target
     - Validasi gagal - Upaya validasi tidak meyakinkan karena kesalahan
     - Tidak divalidasi - Validasi simulasi tidak dilakukan untuk temuan ini. Ini terjadi ketika validasi tidak diaktifkan atau langkah validasi habis sebelum mencapai temuan ini.
   - Terselesaikan — Apakah temuan telah diselesaikan (Yes/No)
   - Terakhir diperbarui - Timestamp dari pembaruan terbaru
5. Perluas bagian Deskripsi untuk membaca:
   - Penjelasan rinci tentang masalah keamanan
   - Bagaimana kerentanan bisa dieksploitasi
   - Dampak potensial pada aplikasi Anda
   - Langkah verifikasi yang dilakukan agen untuk mengkonfirmasi temuan
6. Perluas bagian Lokasi kode untuk melihat:
   - Jalur file tertentu tempat masalah diidentifikasi
   - Penjelasan singkat tentang apa yang ditemukan di setiap lokasi
   - Lencana nomor baris yang menautkan ke kode yang relevan
7. Perluas bagian Bukti untuk meninjau:
   - Kode, konfigurasi, atau pengamatan spesifik yang mendukung temuan
   - Penjelasan singkat mengapa setiap item menunjukkan kerentanan
   - File dan nomor baris yang terpengaruh untuk setiap bukti
8. Perluas bagian perbaikan yang disarankan untuk melihat:
   - Perubahan AWS Security Agent merekomendasikan untuk memulihkan temuan
   - Contoh kode atau konfigurasi untuk setiap perubahan yang disarankan

> **Tip**
>
> Gunakan lokasi kode untuk menavigasi dengan cepat ke file yang terpengaruh di repositori Anda. Setiap lokasi mencakup konteks yang cukup untuk memahami masalah tanpa membaca seluruh file.

Langkah 8: Prioritaskan dan atasi temuan

Validasi dan ambil tindakan atas temuan untuk memulihkan kerentanan dan meningkatkan postur keamanan aplikasi Anda.

Untuk temuan tingkat keparahan tinggi dengan kepercayaan agen tinggi:

1. Tinjau bagian Deskripsi dan Lokasi Kode secara menyeluruh.
2. Gunakan kode Remediate untuk menghasilkan permintaan tarik dengan perbaikan, atau tinjau PR remediasi otomatis jika Anda mengaktifkan opsi itu.
3. Rencanakan peninjauan kode tindak lanjut untuk memverifikasi bahwa perbaikan efektif.

Untuk temuan tingkat keparahan sedang dengan kepercayaan agen tinggi:

1. Prioritaskan berdasarkan toleransi risiko dan konteks bisnis Anda.
2. Sertakan tugas remediasi dalam perencanaan sprint pengembangan reguler Anda.
3. Pertimbangkan apakah beberapa temuan tingkat keparahan sedang bersama-sama menciptakan risiko yang lebih tinggi.

Untuk temuan kepercayaan Sedang atau Rendah:

1. Tinjau bagian Deskripsi dan Lokasi Kode untuk memvalidasi asumsi dan kondisi di mana kerentanan terjadi.
2. Jika kerentanan valid, prioritaskan berdasarkan tingkat keparahan dan toleransi risiko dan pertimbangkan tugas remediasi.

Langkah 9: Lacak kemajuan remediasi

Gunakan antarmuka temuan dan jalankan kembali untuk melacak kerentanan mana yang telah ditangani.

1. Saat Anda menerapkan perbaikan, gunakan temuan Selesaikan untuk menandai temuan seperti yang dibahas.
2. Jalankan tinjauan kode baru untuk memverifikasi bahwa perbaikan menyelesaikan temuan dan tidak menimbulkan masalah baru.
3. Tinjau tabel All runs pada halaman detail tinjauan kode untuk membandingkan temuan di seluruh proses dari waktu ke waktu.

> **Tip**
>
> Setelah menggabungkan permintaan tarik remediasi, mulailah menjalankan peninjauan kode yang sama untuk mengonfirmasi bahwa kerentanan teratasi. Bandingkan jumlah temuan di antara lari untuk melacak kemajuan Anda.

Langkah selanjutnya

Setelah meninjau temuan ulasan kode Anda:

- Prioritaskan temuan tingkat keparahan tinggi dengan keyakinan tinggi untuk remediasi segera
- Gunakan kode Remediate untuk menghasilkan perbaikan otomatis untuk GitHub-sourced temuan (lihat) the section called “Remediasi temuan tinjauan kode”
- Jalankan ulasan kode tambahan setelah menerapkan perbaikan untuk memverifikasi remediasi
- Sesuaikan sumber atau pengaturan peninjauan kode Anda saat basis kode Anda berkembang (lihat) the section called “Aktifkan peninjauan kode”

Remediasi temuan tinjauan kode

Setelah meninjau temuan keamanan dari tinjauan kode, Anda dapat menggunakan AWS Security Agent untuk menghasilkan perbaikan kode untuk temuan di sumber GitHub repositori. Untuk repositori pribadi, AWS Security Agent membuka permintaan tarik dengan perbaikan yang diusulkan. Untuk repositori publik, AWS Security Agent melampirkan perbedaan yang dapat diunduh ke temuan yang dapat Anda terapkan secara lokal. Temuan dari sumber S3 harus diperbaiki secara manual.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Tinjauan kode lengkap dijalankan dengan temuan
- GitHub repositori terhubung sebagai sumber untuk tinjauan kode
- Akses ke aplikasi web AWS Security Agent
- Keakraban dengan arsitektur aplikasi Anda dan persyaratan keamanan

Cara kerja remediasi kode

Saat Anda memicu remediasi kode untuk suatu temuan, AWS Security Agent menganalisis temuan dan lokasi kodenya, lalu menghasilkan perbaikan kode. Bagaimana perbaikan dikirim tergantung pada sumbernya:

- GitHub Repositori pribadi — AWS Security Agent mengirimkan permintaan tarik ke repositori yang relevan. Permintaan tarik mencakup deskripsi masalah keamanan dan perubahan yang dilakukan.
- GitHub Repositori publik — AWS Security Agent melampirkan perbedaan yang disarankan ke temuan sehingga kerentanan tidak diungkapkan sebelum diperbaiki. Anda dapat mengunduh diff dari aplikasi web dan menerapkannya secara lokal dengan. `git apply /path/to/code_remediation_changes.diff`
- Sumber S3 — Remediasi kode tidak tersedia. Gunakan deskripsi temuan, lokasi kode, dan alasan risiko untuk menerapkan perbaikan secara manual.

> **Important**
>
> Permintaan tarik yang dibuat oleh AWS Security Agent dapat dilihat oleh semua pengguna yang telah membaca akses ke repositori. Tinjau perubahan sebelum menggabungkan untuk memastikan perubahan tersebut selaras dengan persyaratan aplikasi Anda.

Remediasi kode otomatis

Jika Anda mengaktifkan remediasi kode otomatis saat membuat tinjauan kode, AWS Security Agent akan membuat perbaikan untuk semua temuan yang memenuhi syarat segera setelah peninjauan selesai. Anda tidak perlu mengambil tindakan tambahan apa pun — permintaan tarik muncul di GitHub repositori pribadi Anda yang terhubung, dan perbedaan muncul pada temuan dari GitHub repositori publik, saat proses selesai.

Remediasi kode manual

Jika remediasi kode otomatis dinonaktifkan, Anda dapat memicu remediasi untuk temuan individu dari GitHub sumber.

1. Arahkan ke tab Temuan dari peninjauan kode yang telah selesai dijalankan.
2. Pilih temuan yang ingin Anda perbaiki.
3. Di panel detail temuan, pilih Remediate code.
4. AWS Security Agent menghasilkan perbaikan kode dan mengirimkan permintaan tarik ke repositori atau melampirkan diff yang dapat diunduh ke temuan, tergantung pada visibilitas repositori.

Tinjau permintaan tarik remediasi

Setelah AWS Security Agent mengirimkan permintaan tarik remediasi ke repositori pribadi GitHub :

1. Arahkan ke GitHub repositori Anda.
2. Temukan permintaan tarik yang dibuat oleh AWS Security Agent.
3. Tinjau perubahan, termasuk:
   - Deskripsi yang menjelaskan temuan dan perbaikan keamanan
   - Perubahan kode mengatasi kerentanan
   - Setiap konteks yang relevan tentang pendekatan remediasi
4. Gabungkan permintaan tarik jika perbaikan sesuai, atau tutup dan terapkan solusi alternatif.

> **Tip**
>
> Setelah menggabungkan permintaan tarik remediasi, mulailah menjalankan peninjauan kode baru untuk memverifikasi bahwa perbaikan menyelesaikan temuan dan tidak menimbulkan masalah keamanan baru.

Terapkan perbedaan remediasi

Untuk temuan dari GitHub repositori publik, terapkan diff yang dapat diunduh secara lokal.

1. Di panel detail temuan, unduh diff dari bagian remediasi Kode.

2. Di klon lokal Anda dari repositori, jalankan. `git apply /path/to/code_remediation_changes.diff`
3. Tinjau perubahan yang diterapkan, uji terhadap aplikasi Anda, dan komit melalui alur kerja pengembangan normal Anda.

Batasan

- Remediasi kode hanya tersedia untuk temuan dari sumber GitHub repositori. Temuan dari sumber S3 harus diperbaiki secara manual.
- AWS Security Agent menghasilkan perbaikan berdasarkan analisis kerentanannya. Tinjau semua perubahan sebelum menggabungkan atau melakukannya untuk memastikan perubahan tersebut sesuai untuk aplikasi Anda.
- Beberapa temuan kompleks mungkin memerlukan intervensi manual di luar perbaikan otomatis.

Langkah selanjutnya

Setelah memulihkan temuan:

- Tinjau dan gabungkan permintaan tarik di GitHub repositori Anda, atau komit perbedaan yang diterapkan melalui alur kerja normal Anda
- Jalankan tinjauan kode baru untuk memverifikasi perbaikan dan memeriksa masalah yang tersisa
- Selesaikan temuan dalam aplikasi web setelah mengonfirmasi remediasi
- Sesuaikan sumber atau pengaturan peninjauan kode sesuai kebutuhan (lihatthe section called “Aktifkan peninjauan kode”)

Jalankan pemindaian kode diferensial dengan S3

Jalankan pemindaian kode diferensial (diff) untuk menganalisis hanya baris yang diubah dalam kode sumber Anda, daripada melakukan pemindaian repositori penuh. Pemindaian diferensial lebih cepat daripada pemindaian penuh dan menghasilkan temuan yang ditargetkan untuk perubahan kode tertentu, menjadikannya ideal untuk validasi pra-penggabungan dalam alur kerja pengembangan.

Tidak seperti tinjauan kode berbasis permintaan tarik yang dipicu secara otomatis oleh peristiwa penyedia kontrol sumber pihak ketiga (lihatthe section called “Tinjau temuan keamanan kode dalam permintaan tarik”), pemindaian diferensial dimulai secara terprogram melalui AWS Security Agent API

atau SDK. Anda mengunggah file diff terpadu ke S3 dan mereferensikannya saat memulai pekerjaan peninjauan kode.

Cara kerja pemindaian diferensial

Pemindaian diferensial menganalisis perubahan kode Anda dalam konteks penuh repositori Anda. Saat Anda membuat sumber peninjauan kode dengan kode sumber yang diunggah ke S3, pemindaian diferensial menggunakan repositori lengkap sebagai konteks sambil memfokuskan temuan secara khusus pada baris yang diubah di diff Anda. Ini memungkinkan pemindaian untuk mengidentifikasi masalah keamanan yang timbul dari bagaimana perubahan Anda berinteraksi dengan kode yang ada — seperti alur otentikasi yang rusak, penanganan data yang tidak aman di seluruh modul, atau perubahan yang mengekspos kerentanan yang ada.

Proses pemindaian:. Anda mengunggah file diff terpadu (output`git diff`) ke bucket S3 yang terhubung ke Ruang Agen Anda. Anda memanggil `StartCodeReviewJob` API dengan `diffSource` parameter yang menunjuk ke lokasi S3 diff Anda. AWS Security Agent hanya menganalisis kode yang diubah dalam perbedaan untuk kerentanan keamanan dan kepatuhan terhadap persyaratan keamanan organisasi Anda. Temuan mencakup garis yang diubah, dengan peringkat keparahan, lokasi kode, dan panduan remediasi.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Ruang Agen dengan peninjauan kode diaktifkan (lihatthe section called “Aktifkan peninjauan kode”)
- Sumber daya peninjauan kode sudah dibuat untuk repositori target, dengan kode sumber lengkap yang diunggah ke bucket S3 terhubung ke Ruang Agen Anda. Repositori lengkap menyediakan konteks yang memungkinkan analisis lebih dalam tentang bagaimana perubahan Anda berinteraksi dengan kode yang ada. Lihat the section called “Buat ulasan kode” petunjuk tentang cara membuat tinjauan kode dan mengunggah kode sumber.
- Bucket S3 yang terhubung ke Ruang Agen Anda
- Izin IAM untuk mengunggah ke bucket S3 dan menelepon `securityagent:StartCodeReviewJob`
- File diff terpadu yang dihasilkan dari perubahan kode Anda (misalnya, output dari) `git diff main..feature-branch`

> **Tip**
>
> Untuk hasil yang paling akurat, pastikan sumber peninjauan kode Anda memiliki versi terbaru dari kode sumber lengkap Anda yang diunggah ke S3. Pemindaian diff menggunakan repositori lengkap ini sebagai konteks untuk memahami arsitektur aplikasi, aliran data, dan kontrol keamanan yang ada saat menganalisis baris yang diubah.

Langkah 1: Buat file diff

Hasilkan perbedaan terpadu yang mewakili perubahan kode yang ingin Anda pindai.

```
git diff main..feature-branch > changes.diff
```

Atau, buat diff untuk perubahan bertahap:

```
git diff --cached > changes.diff
```

> **Tip**
>
> File diff harus dalam format diff terpadu standar. AWS Security Agent mem-parsing jalur file dan mengubah baris dari header diff untuk mencakup analisisnya.

Langkah 2: Unggah perbedaan ke S3

Unggah file diff ke bucket S3 yang terhubung ke Ruang Agen Anda.

```
aws s3 cp changes.diff s3://my-security-agent-bucket/diffs/changes.diff
```

> **Important**
>
> Bucket S3 haruslah yang sudah terhubung ke Ruang Agen Anda. Peran layanan IAM yang terkait dengan Ruang Agen Anda harus memiliki akses baca ke lokasi ini. Lihat the section called “Aktifkan peninjauan kode” petunjuk tentang menghubungkan bucket S3.

Langkah 3: Mulai pekerjaan peninjauan kode diferensial

Panggil `StartCodeReviewJob` API dengan `diffSource` parameter untuk memulai pemindaian diferensial.

Menggunakan AWS CLI

```
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"}'
```

Menggunakan AWS SDK (Python)

```
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']}")
```

Menggunakan AWS SDK () JavaScript/TypeScript

```
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}`);
```

API segera kembali dengan a `codeReviewJobId` dan `status` dari`IN_PROGRESS`.

Langkah 4: Pantau pemindaian

Polling kode meninjau status pekerjaan sampai selesai.

```
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"
```

Pekerjaan berlangsung melalui fase yang sama dengan tinjauan kode lengkap: Preflight → Analisis statis → Finalisasi. Pemindaian diferensial biasanya selesai lebih cepat daripada pemindaian penuh karena mereka menganalisis lebih sedikit baris kode.

Langkah 5: Tinjau temuan

Setelah pekerjaan selesai, buat daftar temuan untuk pekerjaan peninjauan kode.

```
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"
```

Setiap temuan meliputi:

- Deskripsi masalah keamanan
- Tingkat keparahan (Kritis, Tinggi, Sedang, atau Rendah)
- Lokasi kode yang mereferensikan baris tertentu dalam diff
- Alasan risiko menjelaskan dampak potensial
- Panduan remediasi

Untuk informasi lebih lanjut tentang memahami dan bertindak berdasarkan temuan, lihatthe section called “Tinjau temuan dari tinjauan kode”.

Kuota dan batas

- Jumlah maksimum file yang diubah dalam diff: 3.000 file atau 5 MB total ukuran diff
- Temuan dibatasi pada 30 per pemindaian, diprioritaskan berdasarkan tingkat keparahan (Kritikal > Tinggi> Sedang > Rendah)

Langkah selanjutnya

Setelah menjalankan pemindaian diferensial:

- Tinjau temuan dan terapkan panduan remediasi (lihatthe section called “Tinjau temuan dari tinjauan kode”)
- Aktifkan komentar permintaan tarik untuk peninjauan kode otomatis pada setiap permintaan tarik (lihatthe section called “Aktifkan tinjauan kode permintaan tarik untuk GitHub repositori”)
- Jalankan tinjauan kode lengkap secara berkala untuk menangkap masalah di luar perubahan individual (lihatthe section called “Buat ulasan kode”)
- Gunakan integrasi IDE untuk menjalankan pemindaian diferensial langsung dari lingkungan pengembangan Anda (lihat) the section called “Jalankan pemindaian keamanan kode dari IDE Anda”

Jalankan pemindaian keamanan kode dari IDE Anda

Jalankan pemindaian keamanan kode AWS Security Agent langsung dari IDE Anda menggunakan Kiro atau Claude Code. Integrasi IDE memungkinkan Anda memindai kode sumber lokal Anda untuk kerentanan keamanan, menjalankan pemindaian diferensial hanya pada kode yang diubah, dan melakukan tinjauan model ancaman pada dokumen desain — semuanya tanpa meninggalkan lingkungan pengembangan Anda. Temuan muncul di samping kode Anda dengan panduan remediasi, dan Anda dapat menerapkan perbaikan otomatis langsung dari IDE.

Cara kerja integrasi IDE

Integrasi IDE menggunakan server AWS Security Agent MCP (Model Context Protocol) untuk menghubungkan IDE Anda ke layanan AWS Security Agent:

1. Server MCP mengemas kode sumber Anda ke dalam arsip ZIP.
2. Arsip diunggah ke bucket S3 yang disediakan secara otomatis oleh server di akun AWS Anda.

3. Server memanggil AWS Security Agent API untuk memulai pemindaian.
4. AWS Security Agent menganalisis kode untuk kerentanan keamanan dan kepatuhan terhadap persyaratan keamanan organisasi Anda.
5. Temuan dikembalikan ke IDE dengan lokasi kode, deskripsi, peringkat keparahan, dan saran remediasi.

Server MCP menangani semua orkestrasi — penyediaan ruang agen, pembuatan bucket S3, penyiapan peran IAM, pengemasan kode, dan pemungutan suara pindai — sehingga Anda hanya memerlukan kredensi AWS yang dikonfigurasi secara lokal.

> **Note**
>
> Satu-satunya prasyarat adalah kredensi AWS yang dikonfigurasi di lingkungan lokal Anda (misalnya, melalui, `aws configure` AWS SSO, atau variabel lingkungan). Server MCP secara otomatis menyediakan Ruang Agen, peran layanan IAM, dan bucket S3 pada penggunaan pertama.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Kredensyal AWS dikonfigurasi secara lokal dengan izin berikut:
  - `iam:CreateRole`, `iam:PutRolePolicy` (untuk pengaturan satu kali peran layanan)
  - `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`(untuk perbaikan otomatis)
  - `sts:GetCallerIdentity`
- uv diinstal (pelari paket Python)
- Python 3.10 atau yang lebih baru

- Salah satu IDE berikut:
  - Kiro (instal AWS Security Agent Power)
  - Claude Code (instal plugin AWS Security Agent)

Instal server MCP

Kiro

Instal AWS Security Agent Power dari marketplace Kiro. Power menggabungkan konfigurasi server MCP, instruksi kemudi, dan kait siklus hidup untuk saran pemindaian keamanan otomatis.

Atau, konfigurasikan server MCP secara manual di proyek Anda: `.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"
      }
    }
  }
}
```

Kode Claude

Instal plugin AWS Security Agent, yang menyediakan keterampilan untuk penyiapan, pemindaian lengkap, pemindaian diff, ulasan model ancaman, dan remediasi.

Konfigurasikan server MCP di proyek Anda: `.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**
>
> Anda juga dapat menjalankan server MCP melalui Docker jika Anda memilih untuk tidak menginstal Python secara lokal. Lihat dokumentasi server MCP untuk konfigurasi Docker.

First-time penyiapan

Pada penggunaan pertama, server MCP secara otomatis menyediakan sumber daya AWS yang diperlukan:

Sumber daya	Konvensi nama	Tujuan
Agen Ruang	User-chosen atau dibuat secara otomatis	Wadah untuk pemindaian, ulasan, dan pentest
Peran layanan IAM	`SecurityAgentScanRole`	Diasumsikan oleh `securityagent.amazonaws.com` untuk membaca kode yang diunggah
Bucket S3	`security-agent-scans-{account}-{region}`	Menyimpan kode sumber zip (siklus hidup kedaluwarsa otomatis 30 hari)

Untuk memicu penyiapan secara eksplisit, tanyakan IDE Anda:

```
Set up the security agent
```

Penyiapan memverifikasi kredensyal AWS Anda, membuat atau menggunakan kembali Ruang Agen, menyediakan peran IAM dan bucket S3, dan mengonfirmasi kesiapan. Jika Anda sudah memiliki

Ruang Agen dari AWS Management Console, penyiapan akan menunjukkannya dan menanyakan mana yang akan digunakan.

> **Note**
>
> Pengaturan berjalan secara otomatis pada pemindaian pertama Anda jika belum dikonfigurasi. Anda tidak perlu menjalankannya secara terpisah.

Jalankan pemindaian keamanan penuh

Pindai seluruh proyek Anda untuk mencari kerentanan keamanan:

```
Scan this project for security issues
```

IDE:

1. Mengarsipkan kode sumber Anda (tidak termasuk `.git`,`node_modules`, membangun artefak, dan direktori non-esensial lainnya)
2. Mengunggah arsip ke S3
3. Memulai pekerjaan peninjauan kode lengkap
4. Jajak pendapat untuk penyelesaian (biasanya sekitar 1 jam)
5. Menyajikan temuan yang dikelompokkan berdasarkan tingkat keparahan

Memindai kemajuan

Pemindaian penuh biasanya memakan waktu sekitar 1 jam tergantung pada ukuran basis kode. IDE memeriksa kemajuan setiap 5 menit dan memberi tahu Anda saat hasilnya siap. Anda dapat meminta status kapan saja:

```
How's the scan going?
```

Jalankan pemindaian diferensial

Untuk umpan balik yang lebih cepat selama pengembangan, pindai hanya kode yang berubah sejak git ref:

```
Scan my changes against main for security issues
```

Secara default, jika Anda tidak menentukan referensi dasar, pemindaian akan membandingkan perubahan Anda yang tidak berkomitmen. HEAD Anda dapat secara eksplisit menentukan basis yang berbeda:

```
Diff scan my uncommitted changes
```

Pemindaian diferensial mengunggah konteks repositori lengkap dan patch git diff, lalu menjalankan analisis yang hanya berfokus pada baris yang diubah. Hasil biasanya tiba dalam 5-15 menit.

Untuk informasi selengkapnya tentang API pemindaian diff S3, lihat. the section called “Jalankan pemindaian kode diferensial dengan S3”

Jalankan tinjauan model ancaman

Menganalisis dokumen desain untuk perubahan postur keamanan menggunakan metodologi STRIDE:

```
Run a threat model review on my spec
```

IDE mengidentifikasi `design.md` file Anda `requirements.md` dan (biasanya di bawah `.kiro/specs/`), mengunggahnya bersama kode sumber Anda, dan menjalankan analisis model ancaman. Hasil mengidentifikasi:

- Ancaman keamanan dikategorikan oleh STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege)
- Peringkat keparahan untuk setiap ancaman
- Dampak pada aset tertentu dalam basis kode Anda
- Rekomendasi untuk mitigasi

> **Tip**
>
> Jalankan tinjauan model ancaman sebelum membuat tugas implementasi dari spesifikasi. Ini menangkap masalah desain keamanan sebelum kode ditulis.

Tinjau temuan

Ketika pemindaian selesai, temuan muncul di IDE Anda dikelompokkan berdasarkan tingkat keparahan:

```
# 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
```

Laporan terperinci juga ditulis `.security-agent/findings-{scan_id}.md` dengan informasi lengkap untuk setiap temuan termasuk jenis risiko, skor kepercayaan, lokasi kode, dan kode remediasi.

Untuk melihat temuan dari pemindaian sebelumnya:

```
Show my security findings
```

Terapkan perbaikan otomatis

Setelah meninjau temuan, terapkan remediasi otomatis:

```
Fix the security findings
```

IDE bekerja melalui temuan top-down berdasarkan tingkat keparahan (Kritis → Tinggi → Sedang → Rendah), menerapkan perbaikan kode menggunakan panduan remediasi dari setiap temuan. Setelah semua perbaikan diterapkan, secara otomatis menjalankan pemindaian diff verifikasi untuk mengonfirmasi masalah diselesaikan.

Anda juga dapat memperbaiki temuan individu:

```
Fix the SQL injection in user-service.ts
```

> **Important**
>
> Selalu tinjau perbaikan otomatis sebelum melakukan. Verifikasi bahwa perbaikan tidak merusak fungsionalitas yang ada atau memperkenalkan regresi.

Saran pemindaian otomatis (Kiro)

Saat menggunakan Kiro Power, AWS Security Agent dapat secara otomatis menyarankan pemindaian diff setelah Anda melakukan perubahan kode yang sensitif terhadap keamanan. Power memasang hook yang mengevaluasi setiap giliran pengkodean yang telah selesai dan menyarankan pemindaian saat perubahan memengaruhi:

- Otentikasi atau logika otorisasi
- Kriptografi atau penanganan rahasia
- Validasi input atau sanitasi data
- Permintaan jaringan atau integrasi eksternal
- Konfigurasi keamanan (CORS, header, penanganan sesi)

Anda dapat memilih keluar dari saran otomatis kapan saja dengan menghapus. `.kiro/hooks/security-diff-scan-suggester.kiro.hook`

Jenis pemindaian yang tersedia

Jenis pemindaian	Durasi	Kasus penggunaan	Perintah
Pemindaian penuh	Sekitar 1 jam	Tinjauan komprehensif dari seluruh basis kode	“Pindai proyek saya untuk masalah keamanan”
Pemindaian diff	5—15 menit	Kode yang diubah saja (pra-komit, pra-PR)	“Pindai perubahan saya” atau “Pemindaian Diff terhadap main”

Jenis pemindaian	Durasi	Kasus penggunaan	Perintah
Model ancaman	5—30 menit	Dokumen desain (requirements.md, design.md)	“Jalankan tinjauan model ancaman pada spesifikasi saya”

Variabel-variabel lingkungan

Variabel	Deskripsi	Default
`AWS_REGION`	Wilayah AWS untuk panggilan SecurityAgent API	`us-east-1`
`AWS_PROFILE`	Nama profil kredensi AWS	Profil default
`FASTMCP_LOG_LEVEL`	Tingkat log server MCP (DEBUG, INFO, PERINGATAN, KESALAHAN)	`WARNING`

Pemecahan masalah

“Tidak dikonfigurasi. Jalankan pengaturan terlebih dahulu.”

Anda `.security-agent/config.json` hilang atau Ruang Agen tidak ada lagi. Minta IDE untuk menjalankan penyiapan:

```
Set up the security agent
```

Pemindaian gagal dengan AccessDenied

Pastikan kredensyal AWS Anda memiliki izin IAM yang diperlukan yang tercantum di bagian Prasyarat. Izin hilang yang paling umum adalah `iam:CreateRole` (hanya diperlukan untuk pengaturan pertama kali).

Waktu pemindaian habis atau terlalu lama

- Pemindaian penuh pada basis kode besar dapat memakan waktu lebih dari 1 jam

- Gunakan pemindaian diferensial untuk pengembangan berulang — mereka selesai dalam hitungan menit
- Periksa apakah arsip sumber Anda tidak menyertakan direktori yang tidak perlu (server MCP mengecualikan pola umum secara otomatis)

Perbedaan kosong - tidak ada perubahan untuk memindai

Jika Anda menjalankan pemindaian diff tanpa perubahan yang tidak dilakukan, server MCP melaporkan “Tidak ada perubahan vs HEAD” dan tidak memulai pemindaian. Komit atau tahapkan perubahan Anda terlebih dahulu, atau tentukan referensi dasar yang berbeda.

Kuota dan batas

- Ukuran arsip sumber maksimum: 2 GB
- Siklus hidup bucket S3: kode yang diunggah dihapus otomatis setelah 30 hari

Langkah selanjutnya

Setelah menjalankan pemindaian keamanan IDE pertama Anda:

- Aktifkan komentar tinjauan kode permintaan tarik untuk GitHub integrasi otomatis (lihatthe section called “Aktifkan tinjauan kode permintaan tarik untuk GitHub repositori”)
- Mengonfigurasi persyaratan keamanan untuk validasi kebijakan khusus organisasi (lihat) the section called “Kelola persyaratan keamanan”
- Jalankan pemindaian penuh berkala untuk menangkap masalah di seluruh basis kode Anda (lihat) the section called “Buat ulasan kode”
- Gunakan ulasan model ancaman pada spesifikasi fitur baru sebelum implementasi

Buat tes penetrasi

Siapkan pengujian penetrasi otomatis untuk aplikasi web Anda dengan mengonfigurasi cakupan pengujian, domain target, dan akses sumber daya AWS. Tes penetrasi membantu mengidentifikasi kerentanan keamanan dalam menjalankan aplikasi dengan mensimulasikan skenario serangan dunia nyata terhadap domain terverifikasi Anda.

AWS Security Agent melakukan pengujian keamanan komprehensif terhadap aplikasi web Anda berdasarkan cakupan dan izin yang dikonfigurasi, memberikan temuan terperinci tentang kerentanan yang dapat dieksploitasi sebelum penyerang dapat menemukannya.

Dalam prosedur ini, Anda akan membuat pengujian penetrasi dengan mengonfigurasi detail pengujian, menentukan cakupan pengujian, dan menyiapkan izin yang diperlukan.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Akses ke aplikasi web AWS Security Agent
- Setidaknya satu domain terverifikasi untuk pengujian
- Peran IAM dengan izin yang sesuai untuk AWS Security Agent
- Memahami arsitektur aplikasi Anda dan jalur kritis

Mulai membuat tes penetrasi

Arahkan ke halaman pembuatan pengujian penetrasi di Aplikasi Web Agen.

1. Masuk ke aplikasi web AWS Security Agent.
2. Arahkan ke bagian Tes Penetrasi.
3. Pilih Buat tes penetrasi.

> **Tip**
>
> Hanya domain terverifikasi yang dapat dimasukkan dalam tes penetrasi. Minta admin Anda untuk memverifikasi domain di konsol manajemen AWS. Lihat the section called “Aktifkan domain aplikasi untuk pengujian penetrasi”.

Beri nama tes penetrasi Anda

Berikan nama deskriptif yang membantu mengidentifikasi tujuan dan ruang lingkup tes penetrasi ini.

1. Di bidang Penetration test name, masukkan nama deskriptif untuk pengujian penetrasi Anda.

Example

Nama harus dengan jelas mengidentifikasi aplikasi, lingkungan, atau komponen yang sedang diuji. Maksimum 100 karakter.

Konfigurasikan ruang lingkup uji penetrasi

Tentukan domain dan jalur URL mana yang akan diuji, dan konfigurasikan pengecualian opsional untuk mengontrol batas pengujian.

Tambahkan domain target

Tentukan domain terverifikasi yang akan diuji secara aktif untuk kerentanan keamanan.

1. Di bagian cakupan pengujian Penetrasi, cari URL Target.
2. Perluas bagian Domain terverifikasi untuk melihat domain yang tersedia.
3. Di bidang URL target, masukkan URL domain target.

> **Important**
>
> Hanya domain terverifikasi yang dapat diuji. URL harus berada di bawah domain yang telah Anda verifikasi sebelumnya di AWS Security Agent. Sub-domains domain yang diverifikasi tidak memerlukan verifikasi terpisah.

4. Untuk menambahkan beberapa domain target:
   a. Pilih Tambahkan domain.
   b. Masukkan setiap URL domain tambahan.
5. Untuk menghapus domain target, pilih Hapus di sebelah URL domain.

> **Tip**
>
> Untuk hasil terbaik, sertakan semua domain yang merupakan bagian dari alur pengguna aplikasi Anda, termasuk subdomain untuk API, layanan otentikasi, dan pengiriman konten. Sub-domains domain induk yang diverifikasi tidak memerlukan verifikasi terpisah.

Kecualikan jenis risiko (opsional)

Pilih kategori risiko tertentu untuk dikecualikan dari pengujian jika tidak berlaku untuk aplikasi Anda.

1. Temukan bidang Kecualikan jenis risiko.
2. Pilih dropdown untuk melihat jenis risiko yang tersedia.
3. Pilih satu atau lebih jenis risiko untuk dikecualikan dari tes penetrasi.

> **Note**
>
> Tidak termasuk jenis risiko membatasi ruang lingkup pengujian. Hanya kecualikan jenis risiko yang tidak relevan dengan aplikasi Anda atau yang ingin Anda uji secara terpisah.

Tambahkan jalur URL di luar cakupan (opsional)

Tentukan jalur URL yang tidak boleh diuji selama pengujian penetrasi. AWS Security Agent mengecualikan jalur yang ditentukan dan semua jalur yang bersarang di bawahnya. Misalnya, jika Anda menambahkan `https://example.com/admin` sebagai URL di luar cakupan, `https://example.com/admin/tools` juga di luar cakupan.

1. Temukan bagian Out-of-scope URL.
2. Di kolom input Out-of-scope URL, masukkan jalur URL untuk dikecualikan (misalnya, `/admin/delete` atau`/api/reset`).

> **Warning**
>
> Out-of-scope jalur tidak diuji untuk kerentanan. Pastikan Anda mengecualikan hanya jalur yang tidak boleh diakses selama pengujian, seperti operasi destruktif atau fungsi administratif yang sensitif.

3. Untuk menambahkan beberapa jalur di luar cakupan:
   a. Pilih Tambahkan URL.
   b. Masukkan setiap jalur tambahan.
4. Untuk menghapus jalur, pilih Hapus di sebelah jalur.

Tambahkan domain yang dapat diakses (opsional)

Tentukan domain yang diperlukan untuk pengujian tetapi bukan target untuk pengujian kerentanan.

1. Temukan bagian URL yang Dapat Diakses.
2. Di bidang input URL yang dapat diakses, masukkan domain yang seharusnya dapat diakses selama pengujian.

> **Note**
>
> Tambahkan domain yang dapat diakses untuk layanan pihak ketiga (seperti Okta, Auth0, Stripe) yang berada di luar domain target Anda. Ini diperlukan agar AWS Security Agent dapat mengakses URL ini untuk login dan navigasi selama pengujian. AWS Security Agent TIDAK menguji penetrasi domain ini—domain tersebut hanya digunakan untuk tujuan akses. Domain yang dapat diakses tidak memerlukan verifikasi kepemilikan, meskipun domain tersebut berasal dari domain yang berbeda dari target Anda. Hanya domain target yang memerlukan kepemilikan terverifikasi.

3. Untuk menambahkan beberapa domain yang dapat diakses:
   a. Pilih Tambahkan URL.
   b. Masukkan setiap domain tambahan.
4. Untuk menghapus domain, pilih Hapus di sebelah domain.

Tambahkan header HTTP kustom (opsional)

Tentukan header HTTP kustom yang akan ditambahkan ke permintaan apa pun yang dibuat oleh AWS Security Agent selama pengujian penetrasi.

1. Temukan bagian header HTTP Kustom.
2. Di bidang input header HTTP kustom, masukkan nama dan nilai header yang akan dikaitkan dengan permintaan keluar.

> **Note**
>
> Secara default, AWS Security Agent menambahkan header khusus untuk User-Agentdisetel ke securityagent kecuali nilai User-Agentheader khusus yang berbeda ditentukan.

3. Untuk menambahkan beberapa header kustom:
   a. Pilih Tambahkan header.
   b. Masukkan setiap header kustom tambahan.
4. Untuk menghapus header kustom, pilih Hapus di sebelah header.

Konfigurasikan Peran IAM

Pilih peran layanan yang telah dikonfigurasi sebelumnya untuk pengujian penetrasi ini. AWS Security Agent menggunakan model Space-based izin Agen tempat administrator mengonfigurasi peran IAM saat menyiapkan Ruang Agen Anda. Anda memilih dari peran yang sudah dikonfigurasi dan siap digunakan.

1. Di bagian Izin, cari dropdown peran Layanan.
2. Pilih peran IAM yang memberikan AWS Security Agent akses ke sumber daya AWS yang diperlukan.

> **Important**
>
> Peran IAM yang dipilih harus memiliki izin untuk mengakses sumber daya VPC, CloudWatch Log, dan layanan AWS lainnya yang diperlukan untuk pengujian penetrasi. Verifikasi bahwa peran tersebut memiliki hubungan kepercayaan yang benar dengan AWS Security Agent.

3. Temukan CloudWatch dropdown grup log.
4. Pilih grup log tempat log uji penetrasi akan disimpan. (opsional)

> **Note**
>
> Grup CloudWatch log yang dipilih akan menyimpan log terperinci dari eksekusi uji penetrasi, termasuk permintaan yang dibuat, tanggapan yang diterima, dan kerentanan yang ditemukan.
> Jika Anda tidak memilih grup log, grup CloudWatch log baru akan dibuat secara otomatis dengan `/aws/securityagent` awalan untuk menyimpan log uji penetrasi.

Remediasi kode otomatis

Pilih kotak centang Aktifkan remediasi otomatis.

> **Important**
>
> Untuk memulihkan temuan keamanan di repositori kode sumber Anda, AWS Security Agent dapat mengirimkan permintaan tarik ke repositori Anda. Permintaan tarik mungkin terlihat oleh semua pengguna yang memiliki akses baca ke repositori.

Konfigurasikan sumber daya VPC (opsional)

Jika domain target Anda bersifat pribadi dan di-host dalam VPC, konfigurasikan setelan VPC tempat AWS Security Agent harus menjalankan pengujian penetrasi. Langkah ini hanya diperlukan untuk aplikasi yang tidak dapat diakses publik.

> **Note**
>
> Lewati langkah ini jika domain target Anda dapat diakses publik. Konfigurasi VPC hanya diperlukan untuk menguji aplikasi pribadi yang dihosting dalam Amazon Virtual Private Cloud.

Pilih VPC, subnet, dan grup keamanan untuk lingkungan pengujian penetrasi.

1. Di bagian VPC, cari dropdown ID VPC.
2. Pilih VPC tempat domain target Anda di-host.

> **Important**
>
> VPC yang dipilih harus berisi domain target yang Anda tentukan di Langkah 3. Pastikan VPC memiliki perutean dan konfigurasi jaringan yang sesuai untuk memungkinkan AWS Security Agent mengakses aplikasi Anda.

3. Temukan dropdown Subnet.
4. Pilih satu atau lebih subnet di mana tes penetrasi harus dijalankan.

> **Note**
>
> Pilih subnet yang memiliki akses jaringan ke aplikasi target Anda. Tes penetrasi akan dijalankan dari sumber daya yang digunakan dalam subnet ini.

5. Temukan dropdown grup Keamanan.
6. Pilih grup keamanan yang mengontrol akses jaringan untuk uji penetrasi.

> **Important**
>
> Grup keamanan yang dipilih harus mengizinkan lalu lintas keluar ke domain target Anda dan domain apa pun yang dapat diakses. Pastikan aturan kelompok keamanan mengizinkan akses jaringan yang diperlukan untuk pengujian komprehensif.

Konfigurasikan kredensyal otentikasi (opsional)

Jika domain target Anda memerlukan autentikasi, berikan kredensyal untuk memungkinkan AWS Security Agent mengakses area terlindungi aplikasi Anda selama pengujian penetrasi. Langkah ini hanya diperlukan untuk aplikasi yang memerlukan otentikasi pengguna.

> **Note**
>
> Lewati langkah ini jika domain target Anda tidak memerlukan otentikasi atau jika semua area yang ingin diuji dapat diakses publik. Konfigurasikan kredensyal hanya jika Anda memerlukan AWS Security Agent untuk menguji bagian aplikasi yang diautentikasi.

Tambahkan kredensyal

Berikan kredensyal otentikasi yang akan digunakan AWS Security Agent untuk mengakses aplikasi Anda.

1. Di bagian Credential #1, pilih metode input kredenal:
   - Input kredensyal - Masukkan kredensyal Anda langsung ke AWS Security Agent.

- Pengaturan lanjutan - Untuk informasi kredensyal sensitif, gunakan opsi lanjutan seperti AWS Secrets Manager atau fungsi AWS Lambda. Lihat the section called “Memberikan kredensi otentikasi untuk pengujian penetrasi” untuk detail.

> **Tip**
>
> Untuk lingkungan produksi atau kredensyal sensitif, sebaiknya gunakan opsi pengaturan lanjutan untuk mereferensikan kredensyal yang disimpan secara aman di AWS Secrets Manager atau Systems Manager Parameter Store.

Masukkan detail kredensyal

Berikan nama pengguna dan kata sandi untuk akun yang diautentikasi.

1. Di bidang Nama pengguna, masukkan nama pengguna untuk otentikasi.
2. Di bidang Kata Sandi, masukkan kata sandi untuk otentikasi.

> **Important**
>
> Pastikan kredensyal yang Anda berikan memiliki tingkat akses yang sesuai untuk area yang ingin Anda uji. Kredensyal harus mewakili tingkat akses pengguna biasa daripada hak istimewa administratif.

Pilih domain akses

Tentukan domain target mana yang akan menggunakan kredensil ini untuk otentikasi.

1. Di dropdown domain Access, pilih domain tempat kredensyal ini akan digunakan.

> **Note**
>
> Jika Anda memiliki beberapa domain target yang memerlukan kredensyal berbeda, Anda dapat menambahkan set kredensyal tambahan dengan mengklik Tambahkan kredensyal lain setelah menyelesaikan konfigurasi kredensi ini.

Konfigurasikan prompt login agen (opsional)

Berikan instruksi untuk memandu AWS Security Agent melalui proses autentikasi aplikasi Anda.

1. Perluas bagian prompt masuk Agen jika alur otentikasi Anda memerlukan instruksi khusus.
2. Masukkan instruksi yang menjelaskan cara menggunakan kredensyal yang disediakan dalam alur masuk aplikasi Anda.

> **Note**
>
> Prompt login agen memberi tahu agen cara menerapkan kredensyal Anda ke aplikasi Anda. Ini berguna untuk alur otentikasi yang kompleks, proses login multi-langkah, atau aplikasi dengan prosedur login non-standar. Sertakan petunjuk langkah demi langkah seperti “Arahkan ke/login, masukkan nama pengguna di bidang 'Email', masukkan kata sandi, dan pilih 'Masuk'.”

Tambahkan beberapa kredensyal (opsional)

Jika aplikasi Anda memerlukan beberapa set kredensyal atau domain yang berbeda memerlukan autentikasi terpisah, tambahkan set kredensyal tambahan.

1. Setelah menyelesaikan konfigurasi kredensi pertama, pilih Tambahkan kredensi lain.
2. Ulangi langkah konfigurasi kredenal untuk setiap set kredensi tambahan.
3. Untuk menghapus set kredensyal, pilih Hapus di sebelah header kredensyal.

> **Tip**
>
> Konfigurasikan beberapa kredensyal saat menguji peran pengguna yang berbeda, mengakses beberapa domain yang diautentikasi, atau memverifikasi kontrol akses berbasis peran di aplikasi Anda.

Lampirkan sumber daya tambahan (opsional)

Menyediakan sumber daya tambahan untuk membantu AWS Security Agent melakukan pengujian penetrasi yang lebih menyeluruh dan akurat. Sumber daya tambahan dapat mencakup diagram

arsitektur, dokumentasi API, file konfigurasi, GitHub repositori, atau S3-hosted materi yang memberikan konteks tentang aplikasi Anda.

> **Note**
>
> Sumber daya tambahan bersifat opsional tetapi direkomendasikan. Memberikan informasi komprehensif tentang aplikasi Anda membantu memastikan cakupan pengujian menyeluruh, mengurangi kesalahan positif, dan memberikan hasil yang lebih dapat ditindaklanjuti.

Tambahkan sumber daya ke tes penetrasi

Pilih sumber daya yang ada atau unggah file baru yang akan membantu memandu tes penetrasi.

1. Di bagian Sumber daya yang terhubung, Anda dapat:
   - Pilih Pilih dari yang tersedia untuk memilih sumber daya yang sudah terhubung ke AWS Security Agent (seperti GitHub repositori atau bucket S3).
   - Pilih Unggah untuk menambahkan file baru langsung dari sistem lokal Anda.

> **Tip**
>
> Sumber daya yang berguna termasuk dokumentasi API, diagram arsitektur, OpenAPI/Swagger spesifikasi, file konfigurasi, diagram alur otentikasi, dan materi lain yang menggambarkan struktur dan perilaku aplikasi Anda.

Pilih dari sumber daya yang tersedia

Pilih dari sumber daya yang sudah terintegrasi dengan AWS Security Agent.

1. Pilih Pilih dari sumber daya yang ada.
2. Jelajahi daftar sumber daya yang tersedia dari sumber yang terhubung seperti:
   - GitHub repositori, di bawah tab repositori GitHub
   - Bucket S3
   - File yang diunggah sebelumnya
   - Repositori dokumentasi

3. Pilih sumber daya yang ingin Anda sertakan dalam tes penetrasi.
4. Pilih Tambahkan ke uji penetrasi untuk melampirkan sumber daya yang dipilih.

Example

Sebaiknya pilih dan tambahkan GitHub repositori yang relevan ke pentest Anda, sehingga AWS Security Agent dapat mengembangkan pemahaman tentang konteks aplikasi Anda, dan menghasilkan perbaikan kode yang siap diterapkan melalui permintaan tarik (saat diaktifkan)

> **Note**
>
> Sumber daya yang dipilih dari sumber yang tersedia tetap disinkronkan dengan lokasi aslinya. Jika Anda memperbarui file GitHub repositori atau S3, tes penetrasi akan menggunakan versi yang diperbarui.

> **Note**
>
> Jika Anda memiliki VPC pribadi yang terkait dengan pentest dan GitHub repositori yang dikonfigurasi, pastikan itu GitHub dapat diakses melalui VPC pribadi Anda untuk menarik sumber daya. GitHub Dalam kebanyakan kasus, Anda harus memastikan bahwa lalu lintas keluar melalui VPC NAT Gateway diizinkan secara default atau mengonfigurasi aturan khusus untuk mengizinkan lalu lintas keluar GitHub untuk IP (lihat Meta API Endpoint) GitHub

Unggah sumber daya baru

Unggah file langsung dari sistem lokal Anda atau berikan konten teks biasa ke AWS Security Agent.

1. Pilih Unggah.
2. Pilih salah satu metode input berikut:
   - Unggah file lokal - Pilih satu atau beberapa file dari sistem lokal Anda.
   - Tempel teks biasa - Ketik atau tempel konten teks langsung ke bidang input. Pilih Unggah.
3. Kemudian pilih Tambahkan untuk menyelesaikan pengunggahan.
4. Sumber daya yang diunggah muncul di tabel Sumber daya yang terhubung.

> **Tip**
>
> Gunakan opsi teks biasa saat Anda ingin memberikan daftar titik akhir API, pola URL, instruksi pengujian, atau informasi berbasis teks lainnya dengan cepat tanpa membuat file terpisah.

> **Important**
>
> Pastikan file yang diunggah dan konten yang ditempel tidak mengandung informasi sensitif seperti kredensyal produksi, kunci pribadi, atau informasi identitas pribadi (PII). Gunakan versi file konfigurasi dan dokumentasi yang disanitasi.

Connect sumber daya yang ada

Sumber daya yang ada dapat berasal dari apa yang sebelumnya Anda unggah ke AWS Security Agent, dari bucket S3, dan repositori terintegrasi GitHub Anda. Pilih Pilih dari sumber daya yang ada untuk memilihnya.

Kelola sumber daya yang terhubung

Tinjau, atur, dan hapus sumber daya yang melekat pada uji penetrasi.

Tabel sumber daya Terhubung menampilkan semua sumber daya yang termasuk dalam uji penetrasi dengan informasi berikut:

- Nama - Nama file atau pengenal sumber daya
- Jenis - Kategori sumber daya (File yang diunggah, sumber daya S3, GitHub repositori, dll.)

Untuk mengelola sumber daya:

1. Pilih satu atau beberapa sumber daya menggunakan kotak centang.
2. Pilih Hapus dari uji penetrasi untuk melepaskan sumber daya yang dipilih.

> **Note**
>
> Anda dapat mengurutkan tabel berdasarkan Nama atau Jenis dengan mengklik header kolom. Ini membantu mengatur sumber daya saat bekerja dengan banyak file.

Buat tes penetrasi

Selesaikan dan luncurkan konfigurasi pengujian penetrasi Anda.

Setelah mengonfigurasi semua pengaturan, Anda siap untuk membuat tes penetrasi.

1. Tinjau semua bagian konfigurasi untuk memastikan akurasi.
2. Pilih salah satu opsi berikut:
   - Pilih Buat penetrasi untuk menyimpan konfigurasi tanpa segera menjalankannya.
   - Pilih Buat dan jalankan untuk menyimpan konfigurasi dan segera memulai tes penetrasi.
   - Pilih Batal untuk membuang konfigurasi uji penetrasi.

> **Important**
>
> Sebelum menjalankan tes penetrasi, verifikasi bahwa:
>
> - Semua domain target diverifikasi dan dapat diakses dengan benar
> - Peran IAM memiliki izin yang sesuai
> - Out-of-scope jalur dikonfigurasi dengan benar untuk mencegah pengujian operasi destruktif
> - Anda memiliki otorisasi untuk melakukan pengujian keamanan pada semua domain target

> **Note**
>
> Setelah tes penetrasi dimulai, Anda dapat memantau kemajuannya dari bagian Penetration test run. Tes bisa memakan waktu beberapa jam untuk menyelesaikannya. Paling lengkap dalam waktu 16 jam, tergantung pada ruang lingkup dan kompleksitas aplikasi Anda.

Tinjau temuan dari tes penetrasi

Pantau eksekusi pentest secara real time di halaman Penetration Test Logs setelah AWS Security Agent memulai pentest. AWS Security Agent mencatat setiap tindakan selama pentest. Setelah selesai, tinjau ringkasan pentest, yang mencakup ikhtisar aplikasi, cakupan dengan titik akhir yang diidentifikasi, dan penilaian risiko temuan keamanan.

Mengevaluasi temuan keamanan untuk mengatasi kerentanan aplikasi. Setiap temuan berisi penilaian dampak, peringkat keparahan, bukti pendukung, dan detail permintaan tarik remediasi (saat remediasi kode otomatis diaktifkan).

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Uji penetrasi yang sudah selesai atau sedang berlangsung
- Akses ke aplikasi web AWS Security Agent

Langkah 1: Akses uji penetrasi

Arahkan ke proses uji penetrasi Anda untuk melihat halaman ikhtisar, log, dan temuan.

1. Masuk ke aplikasi web AWS Security Agent.
2. Arahkan ke bagian Tes Penetrasi.
3. Pilih uji penetrasi yang ingin Anda periksa dari daftar.

> **Tip**
>
> Halaman detail tes penetrasi menampilkan ringkasan status pengujian, tanggal penyelesaian, dan jumlah temuan yang diidentifikasi.

Langkah 2: Pantau kemajuan tes

Lacak kemajuan uji penetrasi Anda menggunakan indikator langkah.

1. Temukan indikator langkah horizontal di bawah header halaman.

2. Tinjau status setiap fase pengujian:
   - Preflight — Penyiapan awal dan pemeriksaan konektivitas
   - Analisis statis — Analisis kode dan konfigurasi
   - Pentests - Pengujian runtime dan pemindaian kerentanan
   - Finalisasi — Validasi akhir dan pembuatan laporan

> **Note**
>
> Setiap langkah menampilkan indikator status (Selesai, Sedang berlangsung, atau Tertunda). Temuan ditemukan dan divalidasi selama proses pengujian, dengan kerentanan baru muncul saat setiap fase selesai.

Langkah 3: Arahkan ke tab ikhtisar uji penetrasi

1. Bagian Run Summary menyediakan status pengujian, durasi, dan detail tingkat tinggi lainnya. Ini juga menyediakan dasbor temuan keamanan yang dikategorikan berdasarkan tingkat keparahan dan jenis risiko
2. Gambaran umum aplikasi oleh AWS Security Agent memberikan ringkasan uji penetrasi
3. Titik akhir yang ditemukan oleh AWS Security Agent menyediakan daftar semua titik akhir yang ditemukan dan diuji oleh agen AWS Security selama pentest run

Langkah 4: Arahkan ke tab log uji penetrasi

Akses log terperinci dari semua tindakan AWS Security Agent yang dijalankan selama pentest.

1. Tindakan dikategorikan berdasarkan jenis tindakan dan tipe risiko.
2. Pilih tindakan tertentu untuk melihat log terperinci:
   - Ringkasan Pengujian — High-level ringkasan tindakan dan hasil agen
   - Log uji penetrasi - Log terperinci dari semua aktivitas pengujian

> **Note**
>
> Tindakan validator menyediakan log yang memvalidasi temuan di setiap kategori

> **Note**
>
> Tab Temuan menampilkan tampilan terpisah dengan daftar temuan di sebelah kiri dan detail temuan yang dipilih di sebelah kanan.

Langkah 5: Arahkan ke temuan

Setiap temuan dalam daftar menampilkan informasi penting untuk membantu Anda menilai pentingnya dengan cepat.

> **Note**
>
> Filter kepercayaan default
>
> Secara default, Anda hanya melihat temuan dengan kepercayaan agen yang tinggi. Untuk juga menunjukkan temuan dengan kepercayaan agen Sedang atau Rendah dan positif palsu, matikan sakelar Menyembunyikan temuan yang tidak diverifikasi.

Tinjau informasi yang ditampilkan pada setiap kartu pencari:

- Menemukan nama — Judul dan pengenal untuk kerentanan
- Lencana Keyakinan — Menunjukkan tingkat kepercayaan agen dalam temuan (Tinggi, Sedang, atau Rendah)
- Lencana keparahan - Menunjukkan tingkat risiko dengan pengkodean warna:
  - Kritis (merah) — Membutuhkan tindakan segera; eksploitasi dapat menyebabkan kompromi sistem
  - Tinggi (merah) - Membutuhkan perhatian segera; eksploitasi dapat mengakibatkan dampak keamanan yang signifikan
  - Sedang (oranye) - Harus ditangani dalam jangka waktu yang wajar; berkontribusi terhadap risiko keamanan secara keseluruhan
  - Rendah (kuning) - Dapat ditangani sebagai bagian dari perawatan rutin; risiko langsung minimal
  - Informasi (biru) - Untuk tujuan informasi; minimal atau tidak ada risiko langsung
- Stempel waktu pembaruan terakhir - Menunjukkan kapan temuan terakhir diubah atau divalidasi
- Pratinjau deskripsi - Ringkasan singkat tentang kerentanan

> **Important**
>
> Prioritaskan temuan dengan lencana tingkat keparahan kritis atau tinggi dan tingkat kepercayaan tinggi, karena ini mewakili kerentanan yang divalidasi yang membutuhkan perbaikan segera.

Langkah 6: Tinjau detail temuan

Pilih temuan individu untuk melihat informasi komprehensif tentang setiap kerentanan.

1. Pilih nama temuan di panel kiri untuk menampilkan detailnya di panel kanan.
2. Tinjau status validasi:

> **Note**
>
> Jika temuan menampilkan Unknown “Temuan ini belum divalidasi oleh AWS Security Agent,” itu berarti deteksi kerentanan masih dikonfirmasi. Temuan ini mungkin memerlukan verifikasi manual.

3. Tinjau atribut kunci yang ditampilkan di bagian atas:
   - Keyakinan agen — Tingkat kepercayaan AWS Security Agent dalam temuan ini
   - Keparahan — Tingkat risiko dengan lencana kode warna
   - Menemukan log - Pilih “Lacak tindakan & log” untuk melihat log eksekusi rinci dan bukti
   - Jenis risiko — Kategori atau jenis risiko keamanan (misalnya, Bypass Otentikasi, Injeksi SQL)
4. Perluas bagian Deskripsi untuk membaca:
   - Penjelasan rinci tentang kerentanan
   - Bagaimana kerentanan bekerja
   - Mengapa itu mewakili risiko keamanan
   - Dampak potensial pada aplikasi Anda
5. Perluas bagian Penalaran Risiko untuk memahami perhitungan tingkat keparahan:
   - Rincian metrik CVSS (Common Vulnerability Scoring System)
   - Attack Vector (AV) — Bagaimana kerentanan dapat dieksploitasi
   - Attack Complexity (AC) — Betapa sulitnya eksploitasi
   - Privileges Required (PR) - Tingkat akses apa yang dibutuhkan

- Interaksi Pengguna (UI) - Apakah tindakan pengguna diperlukan
- Lingkup (S) — Apakah kerentanan mempengaruhi komponen lain
- Dampak Kerahasiaan, Integritas, dan Ketersediaan

6. Perluas bagian Langkah untuk mereproduksi untuk melihat:
   - Langkah-langkah teknis terperinci untuk menciptakan kembali kerentanan
   - Contoh permintaan dan tanggapan
   - Parameter atau kondisi tertentu yang memicu masalah
7. Bagian Skrip Verifikasi menyediakan cara yang dapat dieksekusi untuk mereproduksi temuan. Perluas bagian ini (bila tersedia) untuk melihat:
   - Petunjuk — Cara mengatur dan menjalankan skrip verifikasi
   - Variabel lingkungan - Daftar variabel yang diperlukan yang harus Anda konfigurasikan sebelum menjalankan skrip. Nilai sensitif disunting untuk keamanan.
   - Unduh Skrip — Pilih untuk mengunduh skrip verifikasi yang dapat dieksekusi

> **Note**
>
> Skrip verifikasi hanya tersedia untuk kerentanan yang dikonfirmasi. Agen harus berhasil menghasilkan dan memvalidasi skrip reproduksi yang dapat dieksekusi.

> **Note**
>
> Skrip verifikasi dibuat menggunakan AI generatif. Tinjau skrip sebelum eksekusi dan jalankan hanya terhadap sistem yang berwenang untuk Anda uji. Untuk panduan tentang AI-generated skrip pengujian secara bertanggung jawab, lihat Kebijakan AI Bertanggung Jawab AWS.

> **Tip**
>
> Skrip verifikasi menyediakan cara yang dapat dieksekusi untuk mereproduksi temuan secara independen. Tetapkan variabel lingkungan yang diperlukan dengan kredensyal Anda sendiri, lalu jalankan skrip terhadap sistem target Anda untuk mengonfirmasi kerentanan ada.

> **Tip**
>
> Gunakan tautan “Lacak tindakan & log” untuk mengakses paket bukti lengkap, termasuk permintaan HTTP, tanggapan, dan upaya eksploitasi yang menunjukkan kerentanan.

Edit temuan

Anda dapat mengedit temuan apa pun untuk memperbaiki detailnya atau menyempurnakan penilaian agen. Bidang berikut dapat diedit:

- Nama — Judul temuan
- deskripsi — Deskripsi rinci tentang kerentanan keamanan
- status — Status temuan saat ini
- RiskType — Jenis risiko keamanan yang diidentifikasi
- RiskLevel - Tingkat keparahan (KRITIS, TINGGI, SEDANG, RENDAH, INFORMASI)
- RiskScore - Skor risiko numerik
- penalaran — Pembenaran untuk skor risiko yang diberikan
- AttackScript — Proof-of-concept kode yang menunjukkan kerentanan
- CustomerNote - Catatan opsional yang menjelaskan alasan Anda untuk mengedit

Untuk mengedit temuan:

1. Arahkan ke halaman detail temuan.
2. Pilih ikon edit untuk memodifikasi salah satu bidang dalam daftar sebelumnya.
3. Simpan perubahan Anda.

> **Note**
>
> Hasil edit Anda disimpan segera dan tercermin dalam temuan. Jika Personalisasi Temuan diaktifkan, semua bidang yang dapat diedit yang Anda ubah akan digunakan untuk menyempurnakan preferensi agen untuk menjalankan masa depan.

Lihat versi agen asli

Saat Anda mengedit temuan, pemilih Versi muncul di header detail pencarian. Ini memungkinkan Anda untuk melihat penilaian agen asli:

1. Temukan pemilih Versi di header detail temuan.
2. Pilih Asli untuk melihat temuan seperti yang awalnya diproduksi oleh agen.
3. Pilih Terbaru untuk kembali ke versi saat ini dengan pengeditan Anda diterapkan.

Tinjau temuan yang dipersonalisasi

Saat Personalisasi Temuan diaktifkan, AWS Security Agent belajar dari hasil edit hingga temuan Anda. Agen menerapkan preferensi tersebut pada temuan serupa dalam uji penetrasi masa depan. Ketika temuan telah disesuaikan berdasarkan suntingan Anda sebelumnya, panel detail menampilkan bagian Perubahan Personalisasi. Untuk informasi tentang mengaktifkan fitur ini, lihat Mengaktifkan atau menonaktifkan Personalisasi Temuan.

1. Temukan bagian Perubahan personalisasi di panel detail pencarian.

> **Note**
>
> Bagian Perubahan Personalisasi hanya muncul pada temuan yang diselaraskan oleh Agen Keamanan AWS dengan hasil edit Anda sebelumnya. Temuan yang tidak cocok dengan preferensi yang dipelajari ditampilkan persis seperti yang dihasilkan agen, tanpa bagian ini.

2. Tinjau ringkasan tentang apa yang berubah dan mengapa. Ringkasan mencantumkan setiap atribut yang disesuaikan dengan nilai asli yang dihasilkan agen, nilai yang dipersonalisasi, dan alasannya. Contoh:

Berdasarkan suntingan Anda sebelumnya, perubahan berikut telah dibuat pada temuan yang awalnya dihasilkan oleh sistem: Tingkat risiko berubah dari MEDIUM menjadi KRITIS; Skor risiko berubah dari 5.3 menjadi 9.5; Penalaran: Tingkat keparahan yang ditingkatkan untuk mencerminkan eksposur penuh kode sumber aplikasi melalui peta sumber yang dapat diakses publik.

1. Tinjau ringkasan untuk memahami bagaimana temuan itu disesuaikan sebelum memutuskan bagaimana menindaklanjutinya.
2. Untuk mengganti temuan yang dipersonalisasi, edit lagi seperti yang Anda lakukan pada temuan lain (misalnya, ubah tingkat keparahan atau skor risiko). Pengeditan terbaru Anda selalu

diutamakan: AWS Security Agent mempelajarinya dan menerapkan preferensi yang diperbarui pada proses berikutnya, menggantikan aturan sebelumnya.

> **Note**
>
> Perubahan personalisasi menyesuaikan atribut pencarian seperti RiskLevel, RiskScore, nama, deskripsi, penalaran, dan AttackScript untuk mencerminkan standar AWS Security Agent yang dipelajari dari hasil edit Anda. Bidang apa pun yang telah Anda edit sebelumnya dapat disesuaikan dengan temuan serupa di masa mendatang.

Bagaimana personalisasi belajar dari suntingan Anda

Saat Personalisasi Temuan diaktifkan, AWS Security Agent belajar dari semua pengeditan yang Anda buat terhadap temuan dan menerapkan penyesuaian serupa untuk proses di masa mendatang. Bidang apa pun yang Anda edit (seperti yang tercantum di bagian Edit temuan sebelumnya) akan digunakan untuk menyempurnakan preferensi yang dipelajari agen.

> **Note**
>
> Untuk bidang status, hanya menandai temuan sebagai FALSE_POSITIVE diperlakukan sebagai preferensi yang dapat dipelajari. Perubahan nilai status lainnya tidak memicu pembelajaran.

Pada pentest berikutnya, agen mengevaluasi temuan baru terhadap preferensi yang Anda pelajari dan menerapkan penyesuaian jika berlaku. Bagian perubahan Personalisasi muncul pada temuan apa pun yang disesuaikan, menjelaskan apa yang diubah.

> **Note**
>
> Personalisasi hanya belajar dari pengeditan yang dilakukan dalam pentest run terbaru yang diselesaikan. Fitur harus diaktifkan pada saat pentest dimulai agar pembelajaran terjadi. Jika Anda mengedit temuan yang sama di kemudian hari, pengeditan terbaru akan diutamakan — agen memperbarui preferensinya untuk mencerminkan keputusan terbaru Anda.

Mengaktifkan atau menonaktifkan Personalisasi Temuan

Personalisasi Temuan diaktifkan secara default. Untuk menonaktifkan atau mengaktifkannya kembali:

1. Arahkan ke konfigurasi pentest Anda.
2. Temukan sakelar Personalisasi Temuan.
3. Untuk mengaktifkan Personalisasi Temuan, aktifkan sakelar. Untuk menonaktifkan, matikan.

Ketika dinonaktifkan, temuan muncul dalam keadaan produksi agen mentah mereka tanpa personalisasi diterapkan. Preferensi yang dipelajari sebelumnya dipertahankan dan akan diterapkan lagi jika fitur diaktifkan kembali.

Langkah 7: Menafsirkan metrik CVSS

Memahami metrik CVSS membantu Anda menilai tingkat keparahan sebenarnya dan memprioritaskan upaya remediasi.

Saat meninjau bagian Penalaran, perhatikan metrik utama ini:

- Attack Vector (Network/Adjacent/Local/Physical) - Menunjukkan seberapa jauh serangan dapat dieksekusi
- Attack Complexity (Low/High) - Menunjukkan apakah kondisi khusus diperlukan untuk mengeksploitasi
- Privileges Required (None/Low/High) - Mengidentifikasi tingkat akses apa yang dibutuhkan penyerang
- Interaksi Pengguna (None/Required) - Menentukan apakah eksploitasi membutuhkan keterlibatan pengguna
- Confidentiality/Integrity/Availability Dampak (None/Low/High) - Mengukur dampak pada keamanan sistem Anda

> **Important**
>
> Temuan dengan vektor serangan Jaringan, Kompleksitas rendah, dan confidentiality/integrity Dampak tinggi mewakili kerentanan paling berbahaya yang membutuhkan perhatian segera.

Langkah 8: Prioritaskan dan atasi temuan

Ambil tindakan atas temuan untuk memulihkan kerentanan dan meningkatkan postur keamanan aplikasi Anda.

Untuk temuan kritis dan tingkat keparahan tinggi dengan keyakinan tinggi:

1. Tinjau Deskripsi dan Langkah-langkah untuk mereproduksi bagian secara menyeluruh.
2. Akses log terperinci melalui tautan “Lacak tindakan & log” untuk mengumpulkan bukti lengkap.
3. Akses perbaikan kode siap implementasi melalui salah satu metode ini:
   - Untuk remediasi otomatis: Gunakan tautan permintaan tarik di bagian remediasi
   - Untuk permintaan manual: Pilih 'Kode Remediasi' di halaman temuan untuk meminta permintaan tarik Prasyarat:
   - Admin harus mengaktifkan remediasi kode untuk GitHub repositori di konsol AWS Security Agent
   - Repositori harus disertakan dalam konfigurasi pentest Anda
4. Rencanakan tes penetrasi tindak lanjut untuk memverifikasi perbaikan efektif.

Untuk temuan tingkat keparahan Sedang dan Rendah:

1. Prioritaskan berdasarkan toleransi risiko dan konteks bisnis Anda.
2. Sertakan tugas remediasi dalam perencanaan sprint pengembangan reguler Anda.
3. Pertimbangkan apakah beberapa temuan tingkat keparahan rendah bersama-sama menciptakan risiko yang lebih tinggi.
4. Dokumentasikan setiap risiko yang diterima dengan pembenaran yang sesuai.

> **Important**
>
> Jangan abaikan temuan tingkat keparahan rendah. Beberapa kerentanan tingkat keparahan rendah seringkali dapat dirantai bersama untuk menciptakan eksploitasi yang lebih serius, terutama bila dikombinasikan dengan rekayasa sosial atau akses fisik.

Langkah 9: Lacak kemajuan remediasi

Gunakan antarmuka temuan untuk melacak kerentanan mana yang telah ditangani dan mana yang memerlukan tindakan lebih lanjut.

1. Saat Anda mengerjakan remediasi, lihat kembali ke bagian Langkah untuk mereproduksi untuk memverifikasi perbaikan Anda.
2. Dokumentasikan pendekatan remediasi Anda untuk setiap temuan untuk referensi masa depan dan audit kepatuhan.

> **Tip**
>
> Pertahankan log remediasi yang memetakan setiap temuan ke resolusinya, termasuk perubahan kode, pembaruan konfigurasi, atau keputusan arsitektur yang dibuat untuk mengatasi kerentanan.

Langkah selanjutnya

Setelah meninjau temuan tes penetrasi Anda:

- Prioritaskan temuan kritis dan tingkat keparahan tinggi dengan keyakinan tinggi untuk remediasi segera
- Unduh, tinjau, dan jalankan skrip verifikasi di lingkungan pengujian Anda untuk menguji skrip dan mengidentifikasi kerentanan
- Buat tiket pelacakan di sistem manajemen masalah Anda dengan tautan untuk menemukan detail dan bukti
- Menerapkan perbaikan dan kontrol keamanan untuk mengatasi kerentanan yang teridentifikasi
- Pantau indikator kemajuan uji penetrasi untuk kerentanan yang baru ditemukan
- Jadwalkan tes penetrasi tindak lanjut untuk memverifikasi bahwa kerentanan telah diperbaiki dengan benar
- Perbarui proses pengujian keamanan aplikasi dan model ancaman berdasarkan temuan
- Tinjau metrik CVSS untuk memahami postur keamanan aplikasi Anda secara keseluruhan

Untuk informasi selengkapnya tentang melakukan tes penetrasi, lihatthe section called “Buat tes penetrasi”.

Untuk informasi selengkapnya tentang memahami siklus hidup Agen Keamanan, lihat. the section called “Memahami hierarki sumber daya dan siklus hidup”

Memberikan kredensi otentikasi untuk pengujian penetrasi

Berikan kredensil untuk mengaktifkan AWS Security Agent menguji area terautentikasi aplikasi web Anda. Tanpa kredensi, agen hanya dapat menguji halaman dan API yang dapat diakses publik.

Konfigurasikan kredensional otentikasi

1. Dalam alur kerja pembuatan pengujian penetrasi, cari bagian Kredensi otentikasi - Opsional.
2. Di bagian Credential #1, pilih metode input kredenal Anda:
   - Input kredensial - Masukkan kredensil secara langsung. Terbaik untuk lingkungan pengembangan dan pengujian.
   - Pengaturan lanjutan - Gunakan manajemen AWS-native kredensi. Direkomendasikan untuk lingkungan produksi dan kredensil sensitif.

Opsi lanjutan

Jika Anda memilih Pengaturan lanjutan, Anda dapat memilih dari tiga strategi kredensi:

- Asumsi peran IAM - Untuk aplikasi yang menggunakan AWS Cognito atau autentikasi IAM
- AWS Secrets Manager - Untuk penyimpanan kredensi yang aman dengan enkripsi dan rotasi
- Fungsi Lambda - Untuk pembuatan kredensi dinamis atau alur otentikasi kompleks

Masukan kredensil secara langsung

1. Pilih Input credentials.
2. Masukkan nama pengguna dan kata sandi.
3. Di menu tarik-turun URL Akses, pilih URL tempat kredensil ini akan digunakan. Ini harus dipilih dari daftar titik akhir target.
4. (Opsional) Di bidang opsional 2FA, berikan rahasia TOTP untuk aplikasi yang memerlukan otentikasi dua faktor. Anda dapat:
   - Masukkan rahasia TOTP secara langsung (misalnya,`JBSWY3DPEHPK3PXP`), atau masukkan `otpauth://totp/` URI lengkap (misalnya,`otpauth://totp/Example:user@example.com?secret=JBSWY3DPEHPK3PXP&issuer=Example`).

- Pilih ikon unggah untuk mengunggah gambar kode QR dari halaman penyiapan aplikasi autentikator Anda. Kode QR dipindai secara lokal dan TOTP URI diekstraksi secara otomatis.

  Ketika rahasia TOTP disediakan, agen secara otomatis menghasilkan kode satu kali baru dan memasukkannya ketika prompt 2FA terdeteksi selama login.

5. (Opsional) Perluas prompt login Ruang Agen untuk memberikan instruksi login tertentu jika aplikasi Anda memiliki alur otentikasi yang kompleks.

> **Important**
>
> Gunakan akun uji dengan akses representatif daripada akun pribadi atau administratif.

Gunakan pengaturan lanjutan

Saat Anda memilih strategi kredensi lanjutan (Secrets Manager, Lambda, atau peran IAM), AWS Security Agent mengambil kredensi Anda langsung dari sumber daya AWS yang dikonfigurasi menggunakan peran layanan. Gunakan prompt login Ruang Agen untuk memberikan petunjuk kepada agen tentang cara menerapkan kredensil tersebut ke aplikasi Anda - misalnya, URL login mana yang akan dinavigasi dan kolom formulir mana yang harus diisi.

> **Note**
>
> Rahasia Secrets Manager dan fungsi Lambda harus berada di akun AWS yang sama dengan penyiapan AWS Security Agent Anda. Cross-account kredensil saat ini tidak didukung.

1. Pilih Pengaturan lanjutan.
2. Di menu tarik-turun strategi akses pengguna, pilih salah satu dari berikut ini:

Pilih peran IAM yang tersedia untuk diasumsikan oleh agen

Gunakan opsi ini untuk aplikasi yang menggunakan AWS Cognito, API Gateway dengan autentikasi IAM, atau sistem otentikasi lainnya. AWS-native Peran IAM harus memiliki hubungan kepercayaan yang memungkinkan AWS Security Agent untuk mengambilnya dan izin untuk mengakses sistem otentikasi aplikasi Anda.

Pilih kredensi statis dari AWS Secrets Manager yang terhubung

Gunakan opsi ini untuk mengambil kredensil dengan aman dari AWS Secrets Manager dengan enkripsi, rotasi, dan audit akses.

Peran IAM harus memiliki `secretsmanager:GetSecretValue` dan `secretsmanager:DescribeSecret` izin.

Agen mengambil nilai rahasia langsung dari Secrets Manager. Gunakan prompt login Ruang Agen untuk memberi tahu agen cara menerapkan kredensil tersebut ke alur masuk aplikasi Anda - misalnya, URL mana yang akan dinavigasi dan kolom formulir mana yang harus diisi. Anda dapat menggunakan format apa pun untuk menyimpan rahasia Anda, karena agen akan menafsirkan format menggunakan instruksi yang Anda berikan dalam prompt login.

Misalnya, jika agen mengirimkan formulir username/password login dihttps://example.com/login, Anda dapat memformat rahasia Anda sebagai JSON dengan `username` dan `password` bidang. Jika aplikasi membutuhkan TOTP-based 2FA, sertakan `totpSecret` bidang dengan rahasia TOTP secara langsung atau URI lengkap: `otpauth://totp/`

```
{
  "username": "test-user",
  "password": "secure-password-here",
  "totpSecret": "JBSWY3DPEHPK3PXP"
}
```

Kemudian, konfigurasikan instruksi otentikasi:. Setel URL Access ke `https://example.com` (atau URL lain yang dipilih dari daftar titik akhir target). Masukkan yang berikut ini ke dalam prompt login Agent Space: “Arahkan ke https://example.com/login dan masukkan nama pengguna dan kata sandi yang disediakan ke dalam formulir.”

Sebagai contoh lain, jika Anda memiliki kunci API yang akan disediakan di header HTTP, Anda dapat menyimpannya sebagai plaintext:

```
"api-key-here"
```

Kemudian, konfigurasikan instruksi otentikasi:. Masukkan yang berikut ini ke prompt login Agent Space: “Setel X-API-Key header ke kunci API yang disediakan untuk semua permintaan.”

> **Important**
>
> Hanya TOTP-based 2FA yang didukung. SMS, email, pemberitahuan push, kunci perangkat keras, dan otentikasi OAuth tidak didukung.

Pilih fungsi Lambda yang tersedia untuk mengambil kredensil secara dinamis

Gunakan opsi ini untuk sistem otentikasi yang kompleks, pembuatan kredensi dinamis, atau integrasi dengan penyedia identitas eksternal.

Peran IAM harus memiliki `lambda:InvokeFunction` izin dan fungsi harus selesai dalam 30 detik.

Saat pengujian penetrasi berjalan, AWS Security Agent memanggil fungsi AWS Lambda Anda secara serempak dan meneruskan peristiwa yang mengidentifikasi pengujian penetrasi dan kredensi spesifik yang sedang diselesaikan:

```
{
  "pentest_arn": "arn:aws:securityagent:us-west-2:111122223333:pentest/pt-
abcd1234-5678-90ab-cdef-EXAMPLE11111",
  "actor_identifier": "Credential1"
}
```

- `pentest_arn`— Nama Sumber Daya Amazon (ARN) dari tes penetrasi yang kredensialnya sedang diselesaikan. Gunakan untuk cakupan, otorisasi, atau audit permintaan dalam fungsi Anda.
- `actor_identifier`— Nama kredenal yang Anda tetapkan untuk kredensi ini di konsol (misalnya,`Credential1`). Gunakan untuk mengembalikan kredensi yang benar ketika satu fungsi menyajikan beberapa kredensil.

Agen menggunakan output fungsi secara langsung sebagai kredensi. Gunakan prompt login Agent Space untuk memberi tahu agen cara menerapkan kredensil tersebut ke aplikasi Anda, seperti yang Anda lakukan dengan AWS Secrets Manager. Lihat the section called “Pilih kredensi statis dari AWS Secrets Manager yang terhubung” contoh cara memformat output fungsi Lambda Anda dan jenis otentikasi yang didukung.

Konfigurasikan beberapa kredensil

Untuk menguji peran pengguna atau sistem otentikasi yang berbeda:

1. Pilih Tambahkan kredensi lain.
2. Konfigurasikan kredensi tambahan menggunakan salah satu metode input.
3. Untuk menghapus kredensi, pilih Hapus di bagian kredensi.

Optimasi Login

Optimasi Login memungkinkan AWS Security Agent untuk mempelajari pola navigasi dari proses pentest sebelumnya dan menerapkannya ke proses berikutnya. Pola-pola ini mencakup alur login dan alur kerja otentikasi multi-langkah. Ini mengurangi waktu yang dihabiskan agen untuk menemukan kembali cara mengotentikasi, menghasilkan pemindaian yang lebih cepat dan cakupan pengujian yang lebih konsisten.

Cara kerja Optimasi Login

Pada pentest run pertama, agen menemukan cara menavigasi alur login aplikasi Anda menggunakan kredensi yang Anda berikan. Ini mencatat langkah-langkah navigasi yang berhasil dan menghasilkan file keterampilan yang menangkap alur kerja login yang dipelajari.

Pada proses berikutnya, agen membaca file keterampilan yang disimpan dan menerapkan pola navigasi yang dipelajari secara langsung, melewatkan fase penemuan. Ini berarti:

- Waktu yang lebih cepat untuk pengujian yang diautentikasi - agen segera menerapkan pola navigasi yang terbukti alih-alih menjelajahi dari awal
- Perilaku yang lebih konsisten — agen mengikuti jalur terbukti yang sama daripada mengeksplorasi alternatif

> **Note**
>
> Optimasi Login belajar selama sesi login pertama dalam proses. Sesi login berikutnya dalam proses yang sama akan mendapat manfaat dari apa yang dipelajari di sesi pertama. Pada masa depan berjalan, semua sesi login mendapat manfaat dari keterampilan yang disimpan.

Aktifkan atau nonaktifkan Optimasi Login

Optimasi Login diaktifkan secara default. Untuk menonaktifkan atau mengaktifkannya kembali:

1. Dalam konfigurasi pentest, arahkan ke bagian kredensi otentikasi - opsional.

2. Temukan sakelar Optimasi Login.
3. Untuk mengaktifkan Optimasi Login, nyalakan sakelar. Untuk menonaktifkan, matikan.

Ketika dinonaktifkan, agen menemukan kembali alur login dari awal pada setiap proses. Keterampilan navigasi yang dipelajari sebelumnya dipertahankan dan akan diterapkan lagi jika fitur diaktifkan kembali.

> **Tip**
>
> Jika alur masuk aplikasi Anda berubah secara signifikan (misalnya halaman login yang didesain ulang atau langkah otentikasi baru), agen secara otomatis beradaptasi dengan memperbarui keterampilan yang dipelajari pada proses berikutnya. Anda tidak perlu mengatur ulang optimasi secara manual.

Memperbaiki temuan tes penetrasi

Saat melihat temuan untuk pengujian penetrasi, Anda dapat meminta AWS Security Agent mencoba memulihkan temuan. Untuk GitHub repositori pribadi, AWS Security Agent membuka permintaan tarik dengan perbaikan yang diusulkan. Untuk repositori publik, remediasi tersedia sebagai file diff yang dapat diunduh yang dapat Anda terapkan secara lokal.

Anda harus mengaktifkan pencarian remediasi di AWS Management Console. (Lihat the section called “Memungkinkan pengguna untuk memulai remediasi uji penetrasi dan temuan tinjauan kode”). Pengguna dapat memulai remediasi untuk temuan tertentu dari AWS Security Agent Web App.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Uji penetrasi yang sudah selesai atau sedang berlangsung
- Akses ke aplikasi web AWS Security Agent
- Keakraban dengan arsitektur aplikasi dan persyaratan keamanan Anda

Langkah 1: Aktifkan atau nonaktifkan remediasi otomatis

Anda dapat mengonfigurasi opsi remediasi kode saat membuat atau memodifikasi tes penetrasi. Jika Anda mengaktifkan remediasi otomatis, AWS Security Agent akan secara otomatis mencoba

memulihkan GitHub repositori terkait jika Agen mengonfirmasi temuan selama pentest. Anda juga dapat memulai remediasi kode secara manual. Dalam tampilan untuk mengedit detail pengujian Penetrasi, di bagian Remediasi kode otomatis, aktifkan atau nonaktifkan remediasi kode.

Langkah 2: Pilih repositori untuk remediasi kode

1. Pilih Berikutnya sampai ke langkah terakhir Sumber belajar tambahan.
2. Pilih Pilih dari sumber daya.
3. Pilih GitHub repositori.
4. Pilih repositori yang Anda inginkan untuk remediasi kode.
5. Simpan tes penetrasi.
6. Anda dapat melihat repositori yang berhasil dikaitkan di bawah tab Sumber daya pembelajaran pengujian Penetrasi.

Langkah 3: Mulai tes penetrasi dan lihat temuan

Jalankan tes penetrasi untuk mendeteksi temuan. Untuk informasi selengkapnya, lihat the section called “Tinjau temuan dari tes penetrasi”.

Langkah 4: Mulai dan lihat remediasi kode

1. Arahkan ke temuan.
2. Jika Anda telah mengaktifkan remediasi kode otomatis, remediasi kode akan dimulai setelah AWS Security Agent mengonfirmasi temuan.
3. Jika Anda ingin memulai remediasi kode secara manual, pilih tombol Remediate code.
4. Di bagian Code Remediation dari temuan, Anda dapat melihat status remediasi kode dan tautan ke permintaan tarik. Jika GitHub repositori bersifat publik, remediasi kode tersedia sebagai file yang dapat diunduh, bukan permintaan tarik. Anda dapat menjalankan `git apply /path/to/code_remediation_changes.diff` untuk menerapkan perubahan ke repositori Anda secara lokal.

Keamanan dan referensi

Panduan keamanan, kuota layanan, dan riwayat dokumen untuk AWS Security Agent.

Keamanan di AWS Security Agent

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS Security Agent di AWS Cloud. Ini termasuk infrastruktur layanan, model AI, dan agen pengujian penetrasi. Third-party Auditor secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari program AWS kepatuhan.
- Keamanan dalam cloud – Tanggung jawab Anda meliputi area-area berikut:
  - Mengelola akses ke AWS Security Agent melalui kebijakan dan izin IAM
  - Melindungi konten yang Anda berikan ke layanan, termasuk dokumen desain, repositori kode, dan URL aplikasi untuk pengujian penetrasi
  - Mengkonfigurasi repositori dan aplikasi mana yang dipantau
  - Meninjau dan bertindak atas temuan keamanan yang disediakan oleh layanan
  - Mengamankan aplikasi Anda berdasarkan panduan remediasi yang diberikan
  - Sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Security Agent.

Pertimbangan Keamanan untuk AWS Security Agent dan pengujian penetrasi berbantuan AI

AWS Security Agent adalah agen perbatasan yang secara proaktif mengamankan aplikasi Anda selama siklus hidup pengembangan di semua lingkungan Anda. Ini melakukan tinjauan keamanan otomatis yang disesuaikan dengan kebutuhan Anda, dengan tim keamanan secara terpusat

mendefinisikan standar yang secara otomatis divalidasi selama peninjauan. Agen Keamanan melakukan pengujian penetrasi sesuai permintaan yang disesuaikan dengan aplikasi Anda, menemukan dan melaporkan risiko keamanan terverifikasi. Pendekatan ini menskalakan keahlian keamanan di seluruh aplikasi Anda agar sesuai dengan kecepatan pengembangan sekaligus memberikan cakupan keamanan yang komprehensif. Dengan mengintegrasikan keamanan dari desain hingga penerapan, ini membantu mencegah kerentanan sejak dini dan dalam skala besar.

Tim keamanan menentukan persyaratan keamanan organisasi sekali di AWS Console: pustaka otorisasi yang disetujui, standar pencatatan, dan kebijakan akses data. AWS Security Agent secara otomatis memberlakukan persyaratan keamanan ini selama pengembangan, mengevaluasi dokumen arsitektur dan kode sesuai standar Anda dan memberikan panduan khusus ketika mendeteksi pelanggaran. Ini memberikan penegakan keamanan yang konsisten di seluruh tim dan tinjauan skala agar sesuai dengan kecepatan pengembangan.

Untuk validasi penerapan, AWS Security Agent mengubah pengujian penetrasi dari bottleneck berkala menjadi kemampuan sesuai permintaan. Tim keamanan menyediakan URL target, detail otentikasi, kode sumber, dan dokumentasi. AWS Security Agent mengembangkan pemahaman aplikasi yang mendalam dan mengeksekusi rantai serangan canggih untuk menemukan dan memvalidasi kerentanan, memungkinkan tim untuk menguji kapan pun diperlukan.

Kemampuan kunci

AWS Security Agent menyediakan kemampuan keamanan komprehensif yang mencakup seluruh siklus hidup pengembangan.

Tinjauan keamanan desain

AWS Security Agent memberikan umpan balik keamanan sesuai permintaan pada dokumen desain dan menilai kepatuhan terhadap persyaratan keamanan organisasi sebelum kode ditulis. Tim keamanan mengunggah dokumen desain melalui aplikasi web, di mana agen menganalisisnya terhadap persyaratan keamanan Anda dan memunculkan temuan dengan panduan remediasi. Ini mempercepat tinjauan manual selama berjam-jam ke dalam analisis terfokus, memungkinkan tim untuk mengatasi masalah keamanan saat remediasi paling efisien.

Tinjauan keamanan kode

AWS Security Agent menganalisis permintaan tarik atau kode yang diunggah untuk persyaratan keamanan organisasi dan masalah keamanan umum seperti validasi input yang hilang dan risiko injeksi SQL. Agen memberikan panduan remediasi langsung dalam platform repositori kode Anda.

Tim keamanan mengonfigurasi repositori mana yang akan dipantau, menskalakan evaluasi di semua basis kode sambil mempertahankan pengawasan pada masalah kritis.

On-demand pengujian penetrasi

AWS Security Agent menyediakan pengujian penetrasi sesuai permintaan yang menemukan dan melaporkan kerentanan keamanan yang divalidasi melalui skenario serangan multi-langkah yang disesuaikan. AWS Security Agent menyebarkan agen AI khusus yang mengembangkan konteks aplikasi dari dokumentasi dan kredensional yang disediakan, kemudian menjalankan rantai serangan canggih untuk mengidentifikasi kerentanan kompleks yang dilewatkan oleh alat konvensional. Ini mendokumentasikan temuan dengan analisis dampak, jalur serangan yang dapat direproduksi, dan perbaikan kode yang siap diterapkan, mempercepat pengujian penetrasi dari minggu ke jam dan validasi penskalaan di seluruh portofolio aplikasi Anda.

FAQ

&Kontrol Keamanan

Bagaimana AWS Security Agent mengautentikasi dan memelihara akses ke sistem?

Pengujian penetrasi adalah satu-satunya kemampuan di AWS Security Agent yang dapat mengautentikasi ke sistem pengguna saat runtime. AWS Security Agent menerima kredensi dalam bentuk kredensi nama pengguna dan kata sandi statis (disimpan di Secrets Manager), atau vendor kredensi (sebagai Fungsi Lambda) sebagai konfigurasi sebelum memulai pengujian pena. Kredensyal ini digunakan untuk menjalankan fungsionalitas normal pengguna system/application melalui siklus hidup tes pena. Kami mendorong pengguna untuk membuat kredensyal baru dengan izin cakupan yang tepat untuk tujuan pentesting.

Dapatkah pengguna mengontrol ruang lingkup dan kedalaman pengujian untuk mencegah dampak sistem yang tidak diinginkan?

AWS Security Agent memungkinkan pelanggan memilih kategori kerentanan tertentu untuk dijelajahi di titik akhir. Pengguna dapat menentukan URL di luar cakupan untuk mencegah AWS Security Agent melakukan pengujian penetrasi terhadap target tersebut. https://docs.aws.amazon.com/securityagent/latest/userguide/perform-penetration-test.html

Bisakah AWS Security Agent sendiri menimbulkan risiko keamanan?

AWS Security Agent diinstruksikan untuk menemukan risiko keamanan, tetapi melakukannya dengan menggunakan muatan berdampak minimal yang disengaja (seperti mengekstrak versi SQL alih-alih

menjatuhkan tabel saat serangan injeksi SQL ditemukan). AWS Security Agent juga terbatas pada pagar pembatas deterministik untuk mencegah perilaku berisiko seperti membuat beban berlebihan terhadap aplikasi target. Sementara pagar pembatas sudah ada, mungkin masih ada interaksi logika bisnis yang tidak disengaja atau tidak jelas, oleh karena itu, kami selalu merekomendasikan melakukan pengujian penetrasi terhadap lingkungan pra-produksi.

Data apa yang dikumpulkan AWS Security Agent dan di mana penyimpanannya?

AWS Security Agent memungkinkan pengguna untuk mengunggah artefak untuk memberikan konteks tentang aplikasi mereka yang sedang diuji. Untuk informasi lebih lanjut tentang perlindungan data, lihatthe section called “Perlindungan data”. AWS Security Agent akan secara otomatis memilih wilayah optimal dalam geografi Anda untuk memproses permintaan inferensi Anda. Ini memaksimalkan sumber daya komputasi yang tersedia, ketersediaan model, dan memberikan pengalaman pelanggan terbaik. Data Anda akan tetap disimpan hanya di wilayah tempat permintaan berasal, namun, permintaan input dan hasil keluaran dapat diproses di luar wilayah tersebut. Semua data akan dikirimkan dienkripsi di seluruh jaringan aman Amazon. Untuk informasi lebih lanjut, lihat Inferensi Lintas Wilayah.

Kontrol apa yang ada untuk memblokir pengujian yang tidak sah terhadap titik akhir?

Titik akhir yang ditetapkan sebagai URL target untuk pentesting akan memerlukan validasi DNS atau validasi HTTP sebagai ukuran kepemilikan. AWS Security Agent akan meminta pelanggan untuk menambahkan data TXT ke DNS endpoint atau mengekspos string validasi pengembalian Rute HTTP sebagai bukti kepemilikan. Hanya setelah menunjukkan bukti kepemilikan, pengguna dapat melanjutkan dengan pentest. Permintaan ke URL di luar target dan URL yang dapat diakses akan diblokir oleh jaringan.

Pelanggan bertanggung jawab untuk memastikan mereka memiliki otorisasi yang tepat untuk menguji semua sistem yang mungkin terpengaruh oleh aktivitas pengujian penetrasi mereka. Semua penggunaan AWS Security Agent harus mematuhi AWS Acceptable Use Policy (https://aws.amazon.com/aup/).

Bagaimana cara pengguna memblokir dan melaporkan penyalahgunaan apa pun menggunakan AWS Security Agent?

AWS Security Agent terus memantau permintaan dan upaya mengakses URL yang berada di luar URL target. Jika penyalahgunaan terdeteksi, seperti mencoba menggunakan AWS Security Agent untuk melakukan pengujian tidak sah pada titik akhir pihak ketiga, setiap pentest yang sedang berlangsung di akun akan dihentikan. Pelanggan dapat menghubungi AWS Support atau tim akun AWS mereka untuk meminta bantuan.

Bisakah AWS Security Agent mengganti alur kerja pengujian pena?

AWS Security Agent bukanlah layanan pengujian penetrasi profesional, dan kami mendorong pengguna untuk mengintegrasikan AWS Security Agent ke dalam alur kerja tinjauan keamanan mereka. AWS Security Agent dapat memberikan aksesibilitas ke pengujian penetrasi sesuai permintaan selama fase pengembangan siklus hidup perangkat lunak ketika terlibat dengan profesional pentesting akan terlalu dini, tidak praktis, atau perlu dievaluasi ulang terlalu sering. Profesional keamanan dapat meninjau temuan dari AWS Security Agent untuk memvalidasinya, menjelaskannya, atau memperluasnya untuk temuan baru (jika ada).

Dapatkah pengguna mengatur kontrol akses berbasis peran (RBAC) untuk anggota tim yang berbeda?

Ya. AWS Security Agent terintegrasi dengan AWS IAM Identity Center, memungkinkan admin mengelola anggota tim yang dapat mengakses aplikasi web AWS Security Agent yang memungkinkan pengguna membuat, mengelola, dan melihat ulasan desain dan pentest.

Kemampuan Pengujian

Jenis kerentanan apa yang dapat dideteksi oleh AWS Security Agent?

AWS Security Agent mendeteksi kerentanan di OWASP Top 10 untuk aplikasi web. AWS Security Agent menyediakan jenis risiko spesifik yang dapat Anda sertakan atau kecualikan dalam pengujian yang diuraikan di bawah ini. Temuan dapat muncul dalam kategori risiko ini atau dari temuan baru yang ditemukan dengan mengikuti petunjuk dari kombinasi kategori risiko ini.

- Unggahan File Sewenang-wenang
  - Unggahan File Arbitrary mengonfirmasi bahwa aplikasi harus dapat menangkis file palsu dan berbahaya dengan cara untuk menjaga aplikasi dan pengguna tetap aman
- Kode Injeksi
  - Code Injection adalah istilah umum untuk jenis serangan yang terdiri dari kode injeksi yang kemudian interpreted/executed oleh aplikasi
- Injeksi Perintah
  - Command injection adalah serangan di mana tujuannya adalah eksekusi perintah arbitrer pada sistem operasi host melalui aplikasi yang rentan
- Cross-Site Skrip (XSS)
  - Cross-Site Serangan Scripting (XSS) adalah jenis injeksi, di mana skrip berbahaya disuntikkan ke situs web yang jinak dan tepercaya

- Referensi Objek Langsung Tidak Aman
  - Referensi Objek Langsung Tidak Aman (IDOR) terjadi ketika aplikasi menyediakan akses langsung ke objek berdasarkan input yang disediakan pengguna
- Kerentanan Token Web JSON
  - JWT adalah sumber kerentanan yang umum, baik dalam bagaimana mereka diimplementasikan dalam aplikasi, dan di pustaka yang mendasarinya
- Inklusi File Lokal
  - Kerentanan Inklusi File memungkinkan penyerang untuk memasukkan file, biasanya mengeksploitasi mekanisme “inklusi file dinamis” yang diterapkan dalam aplikasi target
- Penjelajahan Jalan
  - Path Traversal attack (juga dikenal sebagai directory traversal) bertujuan untuk mengakses file dan direktori yang disimpan di luar folder root web
- Eskalasi Hak Istimewa
  - Eskalasi hak istimewa terjadi ketika pengguna mendapatkan akses ke lebih banyak sumber daya atau fungsionalitas daripada yang biasanya diizinkan, dan peningkatan atau perubahan tersebut seharusnya dicegah oleh aplikasi
- Server-Side Permintaan Pemalsuan (SSRF)
  - Server-Side Request Forgery (SSRF) terjadi ketika penyerang dapat menyalahgunakan fungsionalitas di server untuk membaca atau memperbarui sumber daya internal
- Server-Side Templat Injeksi
  - Kerentanan Injeksi Template Sisi Server (SSTI) terjadi ketika input pengguna disematkan dalam template dengan cara yang tidak aman dan menghasilkan eksekusi kode jarak jauh di server
- SQL Injeksi
  - Serangan injeksi SQL terdiri dari penyisipan atau “injeksi” dari query SQL melalui data input dari klien ke aplikasi
- Entitas Eksternal XML
  - Serangan Entitas Eksternal XMLadalah jenis serangan terhadap aplikasi yang mem-parsing input XHTML. Serangan ini terjadi ketika masukan XHTML yang berisi referensi ke entitas eksternal diproses oleh parser XHTML yang dikonfigurasi dengan lemah

Metode otentikasi apa yang didukung AWS Security Agent?

AWS Security Agent mendukung metode otentikasi umum, termasuk OAuth dan JWT. Lihat informasi yang lebih lengkap dalam dokumentasi .

Bagaimana AWS Security Agent menangani pembatasan tarif dan pencegahan Denial of Service (DOS)?

AWS Security Agent memiliki pagar pembatas untuk mencegahnya mengganggu atau menghapus titik akhir yang sedang diuji, termasuk DOS. Ini memiliki kontrol kecepatan internal untuk mendeteksi dan menangani pola lalu lintas yang tidak terduga.

Bisakah AWS Security Agent menguji API REST dan GraphQL?

Ya, AWS Security Agent dapat menguji titik akhir API. Kami mendorong pelanggan untuk menyediakan dokumentasi API sebagai Sumber Daya Pembelajaran Tambahan yang memungkinkan AWS Security Agent memiliki konteks yang lebih baik tentang bentuk dan fungsionalitas setiap API yang sedang diuji.

Bagaimana pengguna dapat memverifikasi bahwa AWS Security Agent telah mencakup semua logika dan titik akhir aplikasi penting?

AWS Security Agent akan melakukan eksplorasi luas pertama dari aplikasi target dan mencoba untuk menggunakannya secara normal sebelum mencoba eksploitasi apa pun. Ini memungkinkannya untuk membangun pemahaman kerja aplikasi saat runtime dan menemukan logika aplikasi dan titik akhir yang penting. Mengingat sifatnya yang stokastik, AWS Security Agent tidak dijamin untuk menemukan dan menguji semua aplikasi dan titik akhir penting untuk aplikasi target apa pun. Aplikasi web AWS Security Agent memberikan visibilitas ke semua titik akhir yang ditemukan dan tindakan yang diambil dalam log pengujian Penetrasi.

&Keandalan Akurasi

Bagaimana AWS Security Agent memvalidasi temuan sebelum melaporkan?

AWS Security Agent menggunakan validator deterministik untuk membantu memvalidasi temuan yang dilaporkan. Dalam jenis risiko di mana tidak mungkin untuk menggunakan validator deterministik, AWS Security Agent akan secara independen memutar ulang langkah-langkah temuan untuk mendapatkan kepercayaan dalam validitas temuan. AWS Security Agent hanya melaporkan temuan kepercayaan tinggi atau sedang dan menyembunyikan temuan yang belum diverifikasi secara default.

Dapatkah AWS Security Agent beradaptasi dengan logika aplikasi khusus?

AWS Security Agent secara opsional menerima kode sumber, model ancaman, dokumen desain, dan dokumentasi API sebagai Sumber Daya Pembelajaran Tambahan untuk mendapatkan konteks yang diarahkan pengguna pada aplikasi target yang digunakan dalam siklus hidup pentest.

Dapatkah pengguna meninjau metodologi pengujian AWS Security Agent sebelum eksekusi?

Saat ini tidak ada cara untuk melihat pratinjau tindakan AWS Security Agent. Rencana AWS Security Agent bersifat dinamis berdasarkan eksplorasi aplikasi target. Pelanggan dapat memantau AWS Security Agent saat melakukan eksplorasi secara real time dengan mengamati log uji penetrasi. Jika log menunjukkan lintasan yang tidak valid atau tidak diinginkan, pelanggan dapat menghentikan pentest run yang sedang berlangsung.

&Penerapan Integrasi

Apakah AWS Security Agent terintegrasi dengan alat keamanan (SIEM, manajemen kerentanan) atau jaringan pipa? CI/CD

AWS Security Agent tidak berintegrasi dengan alat keamanan atau CI/CD saluran pipa yang ada.

Bagaimana AWS Security Agent menangani konfigurasi khusus lingkungan?

AWS Security Agent dapat dikonfigurasi untuk dijalankan dengan peran IAM tertentu, di dalam VPC, dengan kredensil relevan aplikasi yang ditentukan pelanggan, dan dengan repositori sumber Github sebagai referensi kode sumber ke aplikasi target.

Dapatkah AWS Security Agent berjalan di lingkungan yang memiliki celah udara atau terisolasi?

AWS Security Agent dapat dikonfigurasi untuk memiliki konektivitas ke VPC, termasuk yang tidak memiliki akses internet keluar.

Bisakah beberapa anggota tim menjalankan tes secara bersamaan?

AWS Security Agent mendukung 5 pentest run bersamaan per akun, terlepas dari siapa yang memulai pengujian. Pelanggan dapat membuat maksimal 100 Ruang Agen dan 1.000 proyek Pentest.

Dampak Operasional

Apa dampak kinerja pada sistem yang diuji?

AWS Security Agent memiliki pagar pembatas untuk mencegahnya mengganggu atau menurunkan titik akhir yang sedang diuji. Ini termasuk kontrol kecepatan pada jumlah panggilan yang dapat dilakukan AWS Security Agent ke titik akhir. Sistem atau titik akhir yang diuji harus mengharapkan beberapa peningkatan lalu lintas dan peringatan pemantauan potensial dipicu karena aktivitas uji pena. Rekomendasi kami adalah hanya menjalankan AWS Security Agent atau aktivitas pengujian pena apa pun di lingkungan pra-produksi.

Dapatkah pengguna menjadwalkan atau membatasi AWS Security Agent?

AWS Security Agent tidak memiliki API publik atau kemampuan untuk menjadwalkan pengujian pena berjalan. AWS Security Agent juga tidak menawarkan kontrol konkurensi pada permintaan ke titik akhir target saat memulai uji pena. Jika Agen Keamanan AWS menyebabkan masalah pada titik akhir target, pelanggan dapat menghentikan pentest yang sedang berlangsung.

Berapa durasi tipikal untuk penilaian keamanan lengkap?

Runtime untuk setiap pentest tergantung pada luasnya aplikasi target, dan jenis risiko yang dikonfigurasi untuk dinilai. Sebagian besar pentest berjalan lengkap dalam waktu 16 jam.

Kebijakan AWS yang dikelola untuk Agen Keamanan AWS

Kebijakan terkelola AWS adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. Kebijakan terkelola AWS dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan yang dikelola AWS mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan oleh semua pelanggan AWS. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan terkelola AWS. Jika AWS memperbarui izin yang ditentukan dalam kebijakan terkelola AWS, pembaruan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan terkelola AWS saat layanan AWS baru diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat kebijakan terkelola AWS di Panduan Pengguna IAM.

Kebijakan terkelola AWS: SecurityAgentWebAppAPIPolicy

Memberikan izin untuk berinteraksi dengan Security Agent Web Application API. Kebijakan ini memungkinkan pengguna untuk mengonfigurasi dan menjalankan pengujian penetrasi keamanan otomatis, mengelola eksekusi pengujian, melihat temuan keamanan, dan mengakses sumber daya Agen Keamanan. Kebijakan ini mereferensikan jenis sumber daya Instans Agen lama dan tindakan IAM warisan tertentu.

Detail izin

Kebijakan ini memberikan izin untuk berinteraksi dengan layanan Kontrol Pengujian Keamanan (securityagent: *) untuk:

- Pentest Management: Membuat, memperbarui, menghapus, dan daftar tes penetrasi dan pekerjaan pelaksanaannya
- Temuan Keamanan: Lihat, jelaskan, dan perbarui temuan keamanan dari pengujian yang telah selesai, termasuk konten dan metadata terkait.
- Manajemen Tugas: Daftar dan ambil ulasan kode dan tugas peninjauan dokumentasi
- Resource Discovery: Daftar dan lihat instance agen, artefak, integrasi, dan titik akhir yang ditemukan
- Eksekusi Uji: Mulai dan hentikan eksekusi pentest dengan kemampuan pemantauan waktu nyata
- Remediasi Kode: Mulai remediasi kode otomatis untuk temuan keamanan

Untuk melihat versi terbaru dokumen kebijakan JSON, lihat SecurityAgentWebAppAPIPolicydi Panduan Referensi Kebijakan Terkelola AWS.

Kebijakan terkelola AWS: AWSSecurityAgentWebAppPolicy

Memberikan izin untuk berinteraksi dengan Security Agent Web Application API. Kebijakan ini memungkinkan pengguna untuk mengonfigurasi dan menjalankan pengujian penetrasi keamanan otomatis, mengelola eksekusi pengujian, melihat temuan keamanan, dan mengakses sumber daya Agen Keamanan.

Detail izin

Kebijakan ini memberikan izin untuk berinteraksi dengan layanan Kontrol Pengujian Keamanan (securityagent: *) untuk:

- Pentest Management: Membuat, memperbarui, menghapus, dan daftar tes penetrasi dan pekerjaan mereka
- Temuan Keamanan: Lihat, jelaskan, dan perbarui temuan keamanan dari pengujian yang telah selesai, termasuk konten dan metadata terkait.
- Manajemen Tugas: Daftar dan ambil tinjauan kode dan tugas peninjauan desain
- Resource Discovery: Daftar dan lihat ruang agen, artefak, integrasi, dan titik akhir yang ditemukan
- Eksekusi Uji: Memulai dan menghentikan pekerjaan pentest dengan kemampuan pemantauan waktu nyata
- Remediasi Kode: Mulai remediasi kode otomatis untuk temuan keamanan

Untuk melihat versi terbaru dokumen kebijakan JSON, lihat AWSSecurityAgentWebAppPolicydi Panduan Referensi Kebijakan Terkelola AWS.

AWS Security Agents memperbarui kebijakan yang dikelola AWS

Lihat detail tentang pembaruan kebijakan terkelola AWS untuk Agen Keamanan AWS sejak layanan ini mulai melacak perubahan ini.

Untuk menerima pemberitahuan semua perubahan file sumber ke halaman dokumentasi khusus ini, Anda dapat berlangganan URL berikut dengan pembaca RSS:

Ubah	Deskripsi	Date
Menambahkan izin ke AWSSecurityAgentWebAppPolicy.	Izin tambahan `TargetDomain` dan `DesignReviewFeedback` sumber daya untuk jenis sumber daya baru.	Maret 31, 2026
Menambahkan kebijakan terkelola baru AWSSecurityAgentWebAppPolicy.	Menambahkan kebijakan terkelola `AWSSecurityAgentWebAppPolicy` untuk jenis AgentSpace sumber daya baru dan perubahan nama tindakan IAM.	Februari 9, 2026

Ubah	Deskripsi	Date
Menambahkan izin ke SecurityAgentWebApAPIPolicy.	Ditambahkan `securityagent:StartCodeRemediation` untuk memungkinkan pengguna memulai remediasi kode otomatis untuk temuan keamanan.	Januari 20, 2026
Menambahkan izin ke SecurityAgentWebApAPIPolicy.	Ditambahkan `securityagent:BatchGetSecurityTestContentMetadata` untuk memungkinkan pengguna melihat gambar di konsol.	Desember 5, 2025
AWS Security Agent mulai melacak perubahan.	Agen Keamanan AWS mulai melacak perubahan untuk kebijakan yang dikelola AWS.	Desember 2, 2025

Perlindungan data di AWS Security Agent

Model tanggung jawab bersama AWS berlaku untuk perlindungan data di AWS Security Agent. Sebagaimana dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk menjaga kendali terhadap konten Anda yang di-hosting pada infrastruktur ini. Anda juga bertanggung jawab atas konfigurasi keamanan dan tugas manajemen untuk layanan AWS yang Anda gunakan. Untuk informasi tentang perlindungan data di Eropa, lihat postingan blog Model Tanggung Jawab Bersama AWS dan Peraturan Perlindungan Data Umum (GDPR) di Blog Keamanan AWS. Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungi kredensil akun AWS dan menyiapkan pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.

- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya AWS. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap aktivitas AWS, lihat Bekerja dengan CloudTrail jejak di CloudTrail Panduan Pengguna AWS.
- Gunakan solusi enkripsi AWS, bersama dengan semua kontrol keamanan default dalam layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS Security Agent atau layanan AWS lainnya menggunakan konsol, API, AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi saat diam

AWS Security Agent mengenkripsi semua data saat istirahat menggunakan kunci AWS-managed enkripsi secara default. Hal ini mencakup:

- Dokumen desain dan kode — Semua dokumen desain, repositori kode, dan artefak aplikasi yang Anda berikan untuk tinjauan keamanan dienkripsi menggunakan enkripsi. AES-256
- Temuan keamanan — Semua temuan keamanan, laporan kerentanan, dan rekomendasi remediasi dienkripsi saat istirahat.
- Data konfigurasi — Persyaratan keamanan, kebijakan khusus, dan konfigurasi layanan dienkripsi.
- Log audit — Semua log aktivitas layanan dan jejak audit dienkripsi.

AWS Security Agent menggunakan AWS Key Management Service (AWS KMS) untuk mengelola kunci enkripsi. Anda dapat secara opsional menggunakan kunci yang dikelola pelanggan untuk

mengenkripsi data Anda, memberi Anda kontrol penuh atas kunci enkripsi yang melindungi sumber daya Anda. Untuk informasi selengkapnya, lihat the section called “Kunci yang dikelola pelanggan”.

Enkripsi saat bergerak

AWS Security Agent mengenkripsi semua data dalam perjalanan menggunakan Transport Layer Security (TLS) 1.2 atau yang lebih tinggi. Ini berlaku untuk:

- Komunikasi API — Semua panggilan API antara aplikasi Anda dan AWS Security Agent menggunakan HTTPS dengan enkripsi TLS.
- Akses konsol — Konsol AWS Security Agent diakses melalui HTTPS.
- Koneksi repositori — Koneksi ke GitHub dan repositori kode lainnya menggunakan protokol terenkripsi.
- Komunikasi agen — Semua komunikasi antara layanan AWS Security Agent dan agen pengujian penetrasi menggunakan saluran terenkripsi.

Manajemen kunci

AWS Security Agent menggunakan AWS Key Management Service (AWS KMS) untuk mengelola kunci enkripsi. Secara default, data dienkripsi menggunakan AWS-managed kunci. Anda dapat secara opsional menentukan kunci yang dikelola pelanggan saat membuat sumber daya seperti Ruang Agen dan integrasi. Untuk informasi selengkapnya, lihat the section called “Kunci yang dikelola pelanggan”.

Privasi lalu lintas antarjaringan

AWS Security Agent menggunakan internet publik untuk berkomunikasi dengan penyedia kontrol sumber yang dihosting cloud (GitHub GitLab, Bitbucket) dan Confluence Cloud.

Untuk penyedia yang dihosting sendiri (GitLab Self-Managed, Server GitHub Perusahaan), Anda dapat mengonfigurasi koneksi pribadi menggunakan Amazon VPC Lattice untuk menyimpan semua lalu lintas dalam jaringan AWS. Untuk informasi selengkapnya, lihat the section called “Connect ke kontrol sumber yang dihosting secara pribadi”.

Dalam konfigurasi default, AWS Security Agent menggunakan internet publik untuk menjangkau aplikasi Anda untuk pengujian penetrasi. Anda dapat mengonfigurasi pengujian penetrasi secara opsional untuk menggunakan VPC untuk mengakses aplikasi Anda. Untuk informasi selengkapnya, lihat the section called “Connect agen ke sumber daya VPC pribadi”.

Cross-Region pengolahan data

AWS Security Agent menggunakan inferensi lintas wilayah untuk mengoptimalkan sumber daya komputasi yang tersedia dan ketersediaan model. Bergantung pada Wilayah tempat permintaan berasal, kami mungkin memproses prompt input dan hasil keluaran di Wilayah yang berbeda.

- Di AS Timur (Virginia N.) —`us-east-1`, AS Barat (Oregon) —`us-west-2`, Asia Pasifik (Sydney) —, Asia Pasifik (Tokyo) —`ap-southeast-2`, Eropa (Frankfurt) —`ap-northeast-1`, dan Eropa (Irlandia) —`eu-central-1`, `eu-west-1` AWS Security Agent menggunakan inferensi lintas wilayah geografis. Untuk sebagian besar fitur, pemrosesan data tetap berada dalam batas geografis (seperti AS, UE, Australia, atau Jepang) tempat permintaan tersebut berasal. Untuk Code Remediation, permintaan dari Australia dan Jepang diproses di Uni Eropa. Untuk detail perutean khusus fitur, lihat tabel Inferensi Lintas Wilayah.
- Di Asia Pasifik (Mumbai) —`ap-south-1`, Asia Pasifik (Singapura) —`ap-southeast-1`, dan Amerika Selatan (São Paulo) — `sa-east-1` AWS Security Agent menggunakan inferensi lintas wilayah global. Kami dapat memproses permintaan input dan hasil keluaran di Wilayah AWS komersial mana pun.

Dalam semua kasus, data Anda tetap disimpan hanya di Wilayah tempat permintaan tersebut berasal. Semua data yang dikirimkan selama operasi lintas wilayah tetap berada di jaringan AWS dan tidak melintasi internet publik. Kami mengenkripsi data dalam perjalanan antar Wilayah AWS.

Cross-Region inferensi selalu diaktifkan dan tidak dapat dipilih. Untuk detail tentang Regions mana yang memproses permintaan untuk setiap fitur, lihat bagian Inferensi Lintas Wilayah dithe section called “Praktik terbaik keamanan”.

Penghapusan data

Saat Anda menghapus data dari AWS Security Agent:

- Data ditandai untuk dihapus dan tidak lagi dapat diakses melalui layanan.
- Data dihapus dari semua sistem AWS Security Agent dalam waktu 30 hari.

Untuk menghapus data Anda

1. Di konsol AWS, navigasikan ke AWS Security Agent.
2. Pilih data yang ingin Anda hapus (ulasan keamanan, temuan, atau persyaratan khusus).

3. Pilih Hapus dan konfirmasikan penghapusan.

Manajemen identitas dan akses untuk AWS Security Agent

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. IAM administrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya AWS Security Agent. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di AWS Security Agent.

Pengguna layanan — Jika Anda menggunakan layanan AWS Security Agent untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensyal dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur AWS Security Agent untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda.

Jika Anda tidak dapat mengakses fitur di AWS Security Agent, lihatthe section called “Memecahkan masalah identitas dan akses AWS Security Agent”.

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya AWS Security Agent di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS Security Agent. Tugas Anda adalah menentukan fitur dan sumber daya AWS Security Agent mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari selengkapnya tentang cara perusahaan Anda dapat menggunakan IAM AWS Security Agent, lihatthe section called “Bagaimana AWS Security Agent bekerja dengan IAM”.

IAM administrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke AWS Security Agent. Untuk melihat contoh kebijakan berbasis identitas AWS Security Agent yang dapat Anda gunakan, lihat. IAMthe section called “Contoh kebijakan berbasis identitas AWS Security Agent”

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai pengguna root akun AWS Pengguna IAM, atau dengan

mengambil peran IAM . AWS menyarankan untuk menggunakan peran IAM, dan menghindari penggunaan pengguna root secara langsung.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensyal yang disediakan melalui sumber identitas. AWS IAM Identity Center (IAM Identity Center) pengguna, autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas gabungan. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Akun AWS Anda di Panduan Sign-In Pengguna AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat proses penandatanganan Versi Tanda Tangan 4 di AWS General Reference.

Terlepas dari metode autentikasi yang Anda gunakan, Anda mungkin juga diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat Multi-factor autentikasi di Panduan Pengguna AWS IAM Identity Center (penerus AWS Single Sign-On) dan Menggunakan autentikasi multi-faktor (MFA) di AWS dalam Panduan Pengguna IAM.

Pengguna akar akun AWS

Saat pertama kali membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna akar akun AWS dan diakses dengan cara masuk menggunakan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensyal pengguna root Anda dan gunakan untuk melakukan tugas yang hanya dapat dilakukan oleh pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensi pengguna root di Panduan Referensi Manajemen Akun.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensyal sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apa itu Pusat Identitas IAM? di Panduan Pengguna AWS IAM Identity Center (penerus AWS Single Sign-On).

Pengguna IAM dan kelompok-kelompok

An Pengguna IAMadalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami sarankan untuk mengandalkan kredensyal sementara daripada membuat Pengguna IAM yang memiliki kredensyal jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensyal jangka panjang Pengguna IAM, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

IAM Grup adalah identitas yang menentukan kumpulan. Pengguna IAM Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAmadmins dan memberikan izin grup tersebut untuk mengelola sumber daya. IAM

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat Kapan membuat Pengguna IAM (bukan peran) di Panduan Pengguna IAM.

IAM peran

IAM Peran adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan Pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil IAM peran sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat Menggunakan IAM peran dalam Panduan Pengguna IAM.

IAM peran dengan kredensyal sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, IAM Identity Center mengkorelasikan izin yang disetel ke peran. IAM Untuk informasi tentang set izin, lihat Set izin di Panduan Pengguna AWS IAM Identity Center (penerus AWS Single Sign-On).
- Pengguna IAM Izin sementara — Seorang Pengguna IAM dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Cross-account akses — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun yang berbeda untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan IAM peran dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.
- Cross-service akses — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek Amazon S3. Layanan mungkin melakukan ini menggunakan izin kepala panggilan, menggunakan peran layanan, atau menggunakan peran terkait layanan.
  - Izin utama — Saat Anda menggunakan peran Pengguna IAM atau untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Kebijakan memberikan izin kepada principal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memicu

tindakan lain di layanan yang berbeda. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut.

- Peran layanan — Peran layanan adalah IAM peran yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAM Administrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Service-linked peran — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke suatu. Layanan AWS Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Service-linked peran muncul di Anda Akun AWS dan dimiliki oleh layanan. IAM Administrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

- Aplikasi berjalan pada Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredensional sementara untuk aplikasi yang berjalan pada Amazon EC2 instance dan membuat AWS CLI atau permintaan AWS API. Ini lebih baik untuk menyimpan kunci akses dalam Amazon EC2 instance. Untuk menetapkan AWS peran ke Amazon EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada Amazon EC2 instance untuk mendapatkan kredensyal sementara. Untuk informasi selengkapnya, lihat Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan pada Amazon EC2 instance di Panduan Pengguna IAM.

Untuk mempelajari apakah akan menggunakan IAM peran, lihat Kapan membuat IAM peran (bukan pengguna) di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Setiap IAM entitas (pengguna atau peran) dimulai tanpa izin. Secara default, pengguna tidak dapat melakukan apa pun, bahkan tidak mengubah kata sandi mereka sendiri. Untuk memberikan izin kepada pengguna untuk melakukan sesuatu, administrator harus melampirkan kebijakan izin kepada pengguna. Atau administrator dapat menambahkan pengguna ke grup yang memiliki izin yang dimaksudkan. Ketika administrator memberikan izin untuk grup, semua pengguna dalam grup tersebut akan diberi izin tersebut.

IAM kebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Identity-based kebijakan

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti peran Pengguna IAM, atau grup. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna IAM.

Identity-based Kebijakan selanjutnya dapat dikategorikan sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Memilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Resource-based kebijakan

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya seperti Amazon S3 bucket. Administrator layanan dapat menggunakan kebijakan ini untuk menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu (anggota akun, pengguna, atau peran) pada sumber daya tersebut dan dalam kondisi apa. Resource-based kebijakan adalah kebijakan inline. Tidak ada kebijakan berbasis sumber daya terkelola.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) adalah jenis kebijakan yang mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL mirip dengan

kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON. Amazon S3, AWS WAF, dan Amazon VPC merupakan contoh layanan yang mendukung ACL. Untuk mempelajari lebih lanjut tentang ACL, lihat ikhtisar Access Control List (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batas izin — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (atau peran). IAM Pengguna IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah persimpangan kebijakan berbasis identitas entitas dan batas izinnya. Resource-based kebijakan yang menentukan pengguna atau peran di `Principal` bidang tidak dibatasi oleh batas izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat Batas izin untuk IAM entitas di Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap pengguna akar akun AWS. Untuk informasi selengkapnya tentang Organizations dan SCP, lihat Cara kerja SCP di Panduan Pengguna AWS Organizations.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah persimpangan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS Security Agent bekerja dengan IAM

Sebelum Anda menggunakannya IAM untuk mengelola akses ke AWS Security Agent, pelajari IAM fitur apa saja yang tersedia untuk digunakan dengan AWS Security Agent.

Fitur IAM	Dukungan AWS Security Agent
the section called “Identity-based kebijakan untuk AWS Security Agent”	Ya
the section called “Resource-based kebijakan dalam AWS Security Agent”	Tidak
the section called “Tindakan kebijakan untuk AWS Security Agent”	Ya
the section called “Sumber daya kebijakan untuk AWS Security Agent”	Sebagian
the section called “Kunci kondisi kebijakan untuk AWS Security Agent”	Ya
the section called “Daftar kontrol akses (ACL) di AWS Security Agent”	Tidak
the section called “Attribute-based kontrol akses (ABAC) dengan AWS Security Agent”	Tidak
the section called “Menggunakan kredensyal sementara dengan AWS Security Agent”	Ya
the section called “Meneruskan sesi akses untuk AWS Security Agent”	Ya
the section called “Peran layanan untuk AWS Security Agent”	Tidak
the section called “Service-linked peran untuk AWS Security Agent”	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara Layanan AWS kerja AWS Security Agent dan lainnya IAM, lihat Layanan AWS cara kerjanya IAM di Panduan Pengguna IAM.

Identity-based kebijakan untuk AWS Security Agent

Mendukung kebijakan berbasis identitas: Ya

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti pengguna IAM, grup pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan prinsipal dalam kebijakan berbasis identitas karena berlaku untuk pengguna atau peran yang dilampirkan. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat referensi elemen kebijakan IAM JSON di Panduan Pengguna IAM.

Identity-based contoh kebijakan untuk AWS Security Agent

Untuk melihat contoh kebijakan berbasis identitas AWS Security Agent, lihat. the section called “Contoh kebijakan berbasis identitas AWS Security Agent”

Resource-based kebijakan dalam AWS Security Agent

Mendukung kebijakan berbasis sumber daya: Tidak

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau AWS Services.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Menambahkan principal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan

kepercayaan. Ketika prinsipal dan sumber daya berada di Akun AWS yang berbeda, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Tindakan kebijakan untuk AWS Security Agent

Mendukung tindakan Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, prinsipal mana yang dapat melakukan tindakan di sumber daya apa, dan dalam kondisi apa.

`Action`Elemen kebijakan IAM berbasis identitas menggambarkan tindakan atau tindakan spesifik yang akan diizinkan atau ditolak oleh kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Tindakan ini digunakan dalam kebijakan untuk memberikan izin guna melakukan operasi terkait.

Tindakan kebijakan di AWS Security Agent menggunakan awalan berikut sebelum tindakan:`securityagent:`. Misalnya, untuk memberikan izin kepada seseorang untuk membuat lingkungan dengan operasi AWS Security Agent `CreateEnvironment` API, Anda menyertakan `securityagent:CreateEnvironment` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. AWS Security Agent mendefinisikan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
      "securityagent:action1",
      "securityagent:action2"
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `List`, sertakan tindakan berikut:

```
"Action": "securityagent:List*"
```

Sumber daya kebijakan untuk AWS Security Agent

Mendukung sumber daya kebijakan: Sebagian

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin tingkat sumber daya, seperti operasi daftar, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Beberapa tindakan AWS Security Agent API mendukung banyak sumber daya. Misalnya, beberapa lingkungan dapat direferensikan saat memanggil tindakan `ListEnvironments` API. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma.

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

Misalnya, sumber daya lingkungan AWS Security Agent memiliki ARN berikut:

```
arn:${Partition}:securityagent:${Region}:${Account}:environment/${EnvironmentId}
```

Untuk menentukan lingkungan `my-environment-1` dan `my-environment-2` pernyataan Anda, gunakan contoh ARN berikut:

```
"Resource": [
        "arn:aws:securityagent:us-east-1:123456789012:environment/my-environment-1",
        "arn:aws:securityagent:us-east-1:123456789012:environment/my-environment-2"
```

Untuk menentukan semua lingkungan yang dimiliki akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:securityagent:us-east-1:123456789012:environment/*"
```

Kunci kondisi kebijakan untuk AWS Security Agent

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

`Condition`Elemen (atau `Condition` blok) memungkinkan Anda menentukan kondisi di mana pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan Pengguna IAM izin untuk mengakses sumber daya hanya jika ditandai dengan Pengguna IAM namanya. Untuk informasi selengkapnya, lihat elemen IAM kebijakan: variabel dan tag di Panduan Pengguna IAM.

AWS Security Agent mendefinisikan set kunci kondisinya sendiri dan juga mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Daftar kontrol akses (ACL) di AWS Security Agent

Mendukung ACL: Tidak

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Attribute-based kontrol akses (ABAC) dengan AWS Security Agent

Mendukung ABAC (tag dalam kebijakan): Tidak

Menggunakan kredensyal sementara dengan AWS Security Agent

Mendukung kredensial sementara: Ya

Beberapa AWS Services tidak berfungsi saat Anda masuk menggunakan kredensyal sementara. Untuk informasi tambahan, termasuk Layanan AWS mana yang bekerja dengan kredensi sementara, lihat AWS Services yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensyal sementara jika masuk ke AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, saat Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensyal sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensyal sementara secara manual menggunakan AWS CLI atau AWS API. Anda kemudian dapat menggunakan kredensyal sementara tersebut untuk mengakses AWS. AWS menyarankan agar Anda menghasilkan kredensyal sementara secara dinamis alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Meneruskan sesi akses untuk AWS Security Agent

Mendukung sesi akses terusan (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan di AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari prinsipal yang memanggil Layanan AWS, dikombinasikan dengan Layanan AWS yang meminta untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan Layanan AWS atau sumber daya lainnya untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses terusan.

Peran layanan untuk AWS Security Agent

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan

dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS di Panduan Pengguna IAM.

Service-linked peran untuk AWS Security Agent

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke Layanan AWS. Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Service-linked peran muncul di Akun AWS Anda dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Contoh kebijakan berbasis identitas AWS Security Agent

Secara default, Pengguna IAM dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS Security Agent. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. IAM Administrator harus membuat IAM kebijakan yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya tertentu yang mereka butuhkan. Administrator kemudian harus melampirkan kebijakan tersebut ke grup Pengguna IAM atau yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan menggunakan editor JSON di Panduan Pengguna IAM.

Praktik terbaik kebijakan

Identity-based kebijakan menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya AWS Security Agent di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola atau kebijakan terkelola AWS untuk fungsi pekerjaan di Panduan Pengguna IAM.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan

mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat Kebijakan dan izin IAM di Panduan Pengguna IAM.

- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM JSON: kondisi di Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda untuk memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa IAM kebijakan (JSON) dan praktik terbaik. IAM IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat validasi IAM Access Analyzer kebijakan di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) — Jika Anda memiliki skenario yang mengharuskan Pengguna IAM atau root pengguna di akun Anda, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses MFA-protected API di Panduan Pengguna IAM.

Menggunakan konsol AWS Security Agent

Untuk mengakses konsol AWS Security Agent, prinsipal IAM harus memiliki set izin minimum. Izin ini harus memungkinkan prinsipal untuk membuat daftar dan melihat detail tentang sumber daya AWS Security Agent di situs Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksud untuk prinsipal dengan kebijakan yang dilampirkan padanya.

Untuk memastikan bahwa prinsipal IAM Anda masih dapat menggunakan konsol AWS Security Agent, buat kebijakan dengan nama unik Anda sendiri. Lampirkan kebijakan ke kepala sekolah. Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna dalam Panduan Pengguna IAM:

Untuk detail kebijakan, lihatthe section called “Kebijakan terkelola AWS: SecurityAgentWebAppAPIPolicy”.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang cocok dengan operasi API yang Anda coba lakukan.

Memecahkan masalah identitas dan akses AWS Security Agent

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS Security Agent dan IAM.

AccessDeniedException

Jika Anda menerima `AccessDeniedException` saat memanggil operasi AWS API, kredensyal utama IAM yang Anda gunakan tidak memiliki izin yang diperlukan untuk melakukan panggilan tersebut.

```
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
```

Dalam pesan contoh sebelumnya, pengguna tidak memiliki izin untuk memanggil operasi AWS Security Agent `CreateEnvironment` API. Untuk memberikan izin admin AWS Security Agent kepada prinsipal IAM, lihat. the section called “Contoh kebijakan berbasis identitas AWS Security Agent”

Untuk informasi umum selengkapnya tentang IAM, lihat Mengontrol akses ke sumber daya AWS menggunakan kebijakan di Panduan Pengguna IAM.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Agen Keamanan AWS saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah AWS Security Agent mendukung fitur ini, lihatthe section called “Bagaimana AWS Security Agent bekerja dengan IAM”.

- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke sumber lain Akun AWS yang Anda miliki di Panduan Pengguna IAM. Pengguna IAM
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Memberikan Akses kepada Pengguna yang Diautentikasi Secara Eksternal (Federasi Identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan IAM peran dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Respons insiden

AWS Security Agent adalah layanan pengujian keamanan proaktif yang dirancang untuk mengidentifikasi dan mencegah kerentanan sebelum dapat dieksploitasi. Layanan ini berfokus pada validasi keamanan preventif melalui tinjauan desain, analisis kode, dan pengujian penetrasi daripada deteksi atau respons insiden reaktif.

Deteksi insiden

AWS Security Agent tidak menyediakan kemampuan deteksi insiden. Layanan ini beroperasi sebagai alat pengujian keamanan proaktif yang memvalidasi aplikasi untuk kerentanan selama pengembangan dan sebelum penerapan. Untuk pemantauan keamanan runtime dan deteksi insiden, gunakan layanan seperti Amazon GuardDuty, AWS Security Hub, atau Amazon CloudWatch.

Peringatan insiden

AWS Security Agent tidak menghasilkan peringatan real-time untuk insiden keamanan. Layanan ini memberikan temuan keamanan melalui AWS Console setelah menyelesaikan tinjauan desain, analisis kode, atau keterlibatan pengujian penetrasi. Temuan ini mewakili potensi kerentanan yang ditemukan selama pengujian daripada insiden keamanan aktif.

Remediasi insiden

AWS Security Agent tidak menyediakan remediasi insiden otomatis. Layanan mengidentifikasi kerentanan keamanan dan memberikan panduan remediasi, termasuk:

- Deskripsi terperinci tentang kerentanan yang diidentifikasi
- Jalur eksploitasi yang dapat direproduksi untuk temuan yang divalidasi
- Perbaikan kode khusus dan panduan implementasi
- Analisis dampak untuk masalah yang ditemukan

Tim pengembangan dan keamanan menggunakan panduan ini untuk mengatasi kerentanan secara manual sebelum mencapai lingkungan produksi.

Mendukung kegiatan respon insiden

Meskipun AWS Security Agent tidak dirancang untuk respons insiden, tim keamanan dapat menggunakan layanan ini untuk mendukung aktivitas pasca-insiden:

Validasi kerentanan

: Setelah insiden keamanan, gunakan AWS Security Agent untuk menguji apakah kerentanan serupa ada di aplikasi atau lingkungan lain.

Penilaian postur keamanan

: Melakukan pengujian penetrasi untuk memvalidasi peningkatan keamanan yang diterapkan sebagai bagian dari remediasi insiden.

Analisis akar penyebab

: Gunakan kemampuan tinjauan keamanan kode untuk mengidentifikasi bagaimana kerentanan serupa mungkin ada di basis kode lain.

Validasi kepatuhan untuk AWS Security Agent

Untuk mempelajari apakah layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat layanan AWS dalam Lingkup oleh Program Kepatuhan dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program Kepatuhan AWS.

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi lebih lanjut, lihat Mengunduh Laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA – Daftar layanan yang memenuhi syarat HIPAA. Tidak semua layanan AWS memenuhi syarat HIPAA.
- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Panduan Kepatuhan Pelanggan AWS — Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan Pengembang AWS Config – AWS Config; menilai seberapa baik konfigurasi sumber daya Anda sesuai dengan praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub — Layanan AWS ini memberikan tampilan komprehensif tentang status keamanan Anda dalam AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Layanan AWS ini mendeteksi potensi ancaman terhadap akun AWS, beban kerja, container, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit Manager — Layanan AWS ini membantu Anda terus mengaudit penggunaan AWS Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di Agen Keamanan AWS

> **Note**
>
> AWS Security Agent tersedia di Wilayah berikut: US East (Virginia N.) —`us-east-1`, US West (Oregon) —`us-west-2`, Asia Pasifik (Mumbai) —`ap-south-1`, Asia Pasifik

(Singapura) —`ap-southeast-1`, Asia Pasifik (Sydney) —, Asia Pasifik (Tokyo) —`ap-southeast-2`, Eropa (Frankfurt) —`ap-northeast-1`, Eropa (Irlandia) —`eu-west-1`, `eu-central-1` dan Amerika Selatan (Sao Paulo) —. `sa-east-1` Ini menggunakan inferensi lintas wilayah. Di Asia Pasifik (Mumbai), Asia Pasifik (Singapura), dan Amerika Selatan (São Paulo), AWS Security Agent menggunakan inferensi lintas wilayah global, yang merutekan permintaan inferensi ke Wilayah AWS komersial mana pun. Di semua Wilayah lainnya, ia menggunakan inferensi lintas wilayah geografis, yang membuat pemrosesan data dalam batas-batas geografis tertentu. AWS Security Agent adalah alat yang digunakan selama pengembangan aplikasi Anda, dan tidak boleh digunakan sebagai infrastruktur penting atau yang dihadapi pelanggan.

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung dengan jaringan berlatensi rendah, throughput yang tinggi, dan sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan lebih tersedia, toleran terhadap kegagalan, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau ganda tradisional.

Untuk informasi lebih lanjut tentang Wilayah AWS dan Zona Ketersediaan, lihat Infrastruktur Global AWS.

Keamanan infrastruktur di AWS Security Agent

Sebagai layanan terkelola, AWS Security Agent dilindungi oleh keamanan jaringan global AWS. Untuk informasi tentang layanan keamanan AWS dan cara AWS melindungi infrastruktur, lihat AWS Cloud Security. Untuk mendesain lingkungan AWS Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Keamanan di AWS Well-Architected Framework.

Isolasi jaringan

AWS Security Agent adalah layanan terkelola sepenuhnya yang diakses melalui AWS Console dan AWS Security Agent Web Application. Akses ke layanan dikontrol melalui AWS Identity and Access Management (IAM) atau AWS IAM Identity Center, yang dapat diintegrasikan dengan penyedia identitas Anda.

AWS Security Agent mendukung titik akhir VPC antarmuka yang didukung oleh AWS PrivateLink, memungkinkan Anda mengakses API layanan secara pribadi dari VPC Anda tanpa melintasi internet publik. Untuk informasi selengkapnya, lihat the section called “Titik akhir VPC antarmuka”.

AWS Security Agent memerlukan akses internet untuk melakukan pengujian penetrasi pada aplikasi target dan untuk operasi pesawat kontrol. Layanan menggunakan AWS-managed infrastruktur untuk pengujian dan tidak menyediakan instans EC2 atau sumber daya komputasi lainnya di akun Anda. Jika Anda mengonfigurasi akses VPC untuk pengujian penetrasi, Agen Keamanan membuat ENI di subnet Anda, tetapi ENI ini tidak memiliki alamat IP publik. Untuk informasi selengkapnya, lihat the section called “Connect agen ke sumber daya VPC pribadi”.

Multi-tenancy dan isolasi sumber daya

AWS Security Agent adalah layanan multi-tenant. Tinjauan keamanan, temuan, dan data pelanggan diisolasi ke akun AWS individual dan dienkripsi saat istirahat. AWS menerapkan kontrol isolasi infrastruktur standar untuk memastikan bahwa aktivitas pengujian keamanan satu pelanggan tidak memengaruhi kinerja atau kerahasiaan pelanggan lain.

AWS Agen Keamanan dan titik akhir VPC antarmuka (AWS PrivateLink)

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC dan Agen AWS Keamanan Anda. Anda dapat mengakses Agen Keamanan seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi Direct Connect. Instans di VPC Anda tidak memerlukan alamat IP publik untuk mengakses Agen Keamanan.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan untuk Agen Keamanan.

Untuk informasi selengkapnya, lihat Akses AWS layanan melalui AWS PrivateLink AWS PrivateLink Panduan.

Pertimbangan untuk Agen Keamanan

Sebelum Anda menyiapkan titik akhir antarmuka untuk Agen Keamanan, tinjau Pertimbangan dalam Panduan. AWS PrivateLink

Agen Keamanan mendukung panggilan ke semua tindakan API-nya dari VPC Anda, termasuk operasi untuk mengelola ruang agen, pengujian penetrasi, dan temuan keamanan.

Anda dapat memilih IPv4, IPv6, atau dualstack saat membuat endpoint.

Kebijakan titik akhir VPC didukung untuk Agen Keamanan. Secara default, akses penuh ke Agen Keamanan diizinkan melalui titik akhir antarmuka. Anda dapat mengontrol akses dengan melampirkan kebijakan titik akhir ke titik akhir antarmuka atau dengan mengaitkan grup keamanan dengan antarmuka jaringan titik akhir.

Semua panggilan API ke Agen Keamanan dienkripsi menggunakan TLS 1.2 atau lebih tinggi, termasuk panggilan yang dilakukan melalui titik akhir VPC. Untuk informasi selengkapnya tentang perlindungan data, lihatthe section called “Perlindungan data”. Panggilan API Agen Keamanan masuk AWS CloudTrail. Untuk informasi selengkapnya, lihat the section called “Contoh: Entri file log Agen AWS Keamanan”.

Buat titik akhir antarmuka untuk Agen Keamanan

Anda dapat membuat titik akhir antarmuka untuk Agen Keamanan menggunakan konsol Amazon VPC atau Antarmuka Baris AWS Perintah (AWS CLI). Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLink Panduan.

Buat endpoint antarmuka untuk Agen Keamanan menggunakan nama layanan berikut:

```
com.amazonaws.<region>.securityagent
```

Untuk membuat endpoint antarmuka menggunakan AWS CLI, jalankan perintah berikut. Ganti Wilayah, ID VPC, ID subnet, dan ID grup keamanan dengan nilai Anda sendiri.

```
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**
>
> DNS pribadi harus diaktifkan untuk titik akhir antarmuka. Agen Keamanan mengharuskan Anda menggunakan nama DNS Regional default (misalnya,`securityagent.us-`

`east-1.api.aws`) untuk membuat permintaan API melalui titik akhir. Memanggil URL VPCE khusus titik akhir secara langsung tidak didukung.

Untuk menggunakan DNS pribadi, Anda harus mengatur `enableDnsHostnames` atribut `enableDnsSupport` dan atribut `true` untuk VPC Anda. Untuk informasi selengkapnya, lihat Atribut DNS untuk VPC Anda dalam Panduan Pengguna Amazon VPC.

Konfigurasikan grup keamanan untuk titik akhir antarmuka

Saat Anda membuat titik akhir antarmuka, Anda dapat mengaitkan grup keamanan dengan antarmuka jaringan titik akhir untuk mengontrol lalu lintas ke Agen Keamanan. Buat grup keamanan yang memungkinkan lalu lintas HTTPS masuk (port 443) dari sumber daya di VPC Anda yang perlu berkomunikasi dengan Agen Keamanan.

Grup keamanan harus menyertakan aturan masuk berikut:

- Jenis: HTTPS
- Protokol: TCP
- Rentang port: 443
- Sumber: Blok CIDR VPC Anda (misalnya,`10.0.0.0/16`) atau ID grup keamanan sumber daya yang memerlukan akses ke Agen Keamanan

Untuk informasi selengkapnya, lihat Grup keamanan di AWS PrivateLink Panduan.

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah sumber daya IAM yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh ke Agen Keamanan melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan ke Agen Keamanan dari VPC Anda, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (AWS akun, pengguna IAM, dan peran IAM).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan menggunakan kebijakan titik akhir di AWS PrivateLink Panduan.

Contoh: Kebijakan titik akhir VPC untuk AWS tindakan Agen Keamanan

Berikut ini adalah contoh kebijakan endpoint untuk Agen AWS Keamanan. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke tindakan Agen AWS Keamanan yang terdaftar untuk semua prinsipal di semua sumber daya.

```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "securityagent:CreatePentest",
            "securityagent:ListPentests",
            "securityagent:BatchGetPentests",
            "securityagent:StartPentestExecution",
            "securityagent:StopPentestExecution",
            "securityagent:ListFindings",
            "securityagent:BatchGetFindings"
         ],
         "Resource": "*"
      }
   ]
}
```

Contoh: Kebijakan titik akhir VPC yang menolak semua akses dari akun tertentu AWS

Kebijakan titik akhir VPC berikut menyangkal `123456789012` semua akses AWS akun ke sumber daya menggunakan titik akhir. Kebijakan ini mengizinkan semua tindakan dari akun lainnya.

```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "securityagent:*",
         "Resource": "*"
      },
      {
```

```
         "Effect": "Deny",
         "Principal": {
            "AWS": "123456789012"
         },
         "Action": "securityagent:*",
         "Resource": "*"
      }
   ]
}
```

Pemecahan masalah

Waktu koneksi habis saat memanggil Security Agent API

- Verifikasi bahwa status titik akhir VPC adalah: `available`

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1a2b3c4d \
    --query "VpcEndpoints[].State"
```

- Konfirmasikan bahwa grup keamanan yang terkait dengan titik akhir memungkinkan lalu lintas TCP masuk pada port 443 dari CIDR VPC atau grup keamanan sumber Anda.
- Periksa apakah tabel rute VPC Anda tidak merutekan lalu lintas Agen Keamanan ke gateway internet atau gateway NAT alih-alih titik akhir.

Resolusi DNS gagal untuk **`securityagent.<region>.api.aws`**

- Verifikasi bahwa DNS pribadi diaktifkan pada titik akhir:

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1a2b3c4d \
    --query "VpcEndpoints[].PrivateDnsEnabled"
```

- Konfirmasikan bahwa `enableDnsHostnames` keduanya `enableDnsSupport` dan disetel ke `true` pada VPC Anda:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-1a2b3c4d --attribute enableDnsSupport
aws ec2 describe-vpc-attribute --vpc-id vpc-1a2b3c4d --attribute enableDnsHostnames
```

`Access denied`kesalahan saat memanggil API Agen Keamanan

- Periksa kebijakan titik akhir VPC untuk memastikannya memungkinkan tindakan yang Anda coba lakukan.
- Verifikasi bahwa kebijakan identitas IAM Anda memberikan izin yang diperlukan`securityagent:`.
- Jika menggunakan kebijakan titik akhir kustom, konfirmasikan bahwa kebijakan titik akhir dan kebijakan IAM mengizinkan permintaan tersebut.

Analisis konfigurasi dan kerentanan di AWS Security Agent

Konfigurasi dan kontrol IT merupakan tanggung jawab bersama antara AWS dan Anda, pelanggan kami. Untuk informasi selengkapnya, lihat model tanggung jawab bersama AWS.

Cross-service pencegahan wakil bingung

Masalah deputi yang membingungkan adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Di AWS, peniruan identitas lintas layanan dapat mengakibatkan masalah deputi yang membingungkan. Cross-service peniruan identitas dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberi akses ke sumber daya di akun Anda. Untuk informasi selengkapnya, lihat pencegahan deputi yang Cross-service membingungkan di Panduan Pengguna AWS Identity and Access Management.

Praktik terbaik keamanan untuk AWS Security Agent

AWS Security Agent menyediakan sejumlah fitur keamanan untuk dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan sebagai resep.

Gunakan lingkungan non-produksi untuk pengujian penetrasi

AWS Security Agent menggunakan rangkaian alat pengujian penetrasi yang komprehensif dari distribusi Kali Linux. Alat-alat ini dirancang untuk mengidentifikasi kerentanan keamanan dan dapat melakukan tindakan yang mengubah status aplikasi, data, atau konfigurasi sistem.

Praktik terbaik: Lakukan tes penetrasi terhadap lingkungan non-produksi yang mencerminkan pengaturan produksi Anda. Lingkungan pengujian ini harus:

- Tidak mengandung data pelanggan langsung atau informasi produksi sensitif
- Diisolasi dari sistem produksi
- Memiliki konfigurasi dan kontrol keamanan yang serupa dengan produksi
- Tidak menggunakan kredensyal dengan akses ke sistem produksi

Pengujian di lingkungan produksi dapat menghasilkan:

- Modifikasi atau penghapusan data
- Gangguan layanan atau penurunan kinerja
- Perubahan status yang tidak diinginkan
- Memicu peringatan keamanan atau prosedur respons insiden

Validasi temuan AI-generated keamanan

AWS Security Agent melakukan analisis keamanan menggunakan agen AI. Karena sifat non-deterministik sistem AI, uji penetrasi dapat menghasilkan hasil yang bervariasi di seluruh eksekusi yang berbeda.

Praktik terbaik: Validasi temuan keamanan sebelum mengambil tindakan remediasi:

- Tinjau skrip verifikasi yang dihasilkan oleh AWS Security Agent untuk setiap temuan
- Jalankan skrip verifikasi di lingkungan pengujian Anda untuk mengonfirmasi kerentanan
- Pertimbangkan untuk menjalankan beberapa tes penetrasi untuk memastikan cakupan yang komprehensif
- Menerapkan penilaian keamanan profesional untuk menilai tingkat keparahan dan eksploitasi

Tidak semua masalah yang diidentifikasi dapat mewakili kerentanan yang dapat dieksploitasi dalam konteks penerapan spesifik Anda.

Tinjau dan uji kode remediasi yang dihasilkan

AWS Security Agent dapat menghasilkan perbaikan kode dan peningkatan keamanan untuk kerentanan yang teridentifikasi. AI-generated Perbaikan ini memerlukan verifikasi sebelum penerapan.

Praktik terbaik: Tinjau semua perubahan kode yang dihasilkan:

- Periksa perbaikan yang diusulkan untuk kelengkapan dan kebenaran
- Uji perbaikan secara menyeluruh di lingkungan non-produksi
- Verifikasi bahwa perbaikan tidak memperkenalkan kerentanan baru atau merusak fungsionalitas
- Gunakan AWS Security Agent untuk menguji ulang setelah menerapkan perbaikan, atau menjalankan skrip verifikasi yang disediakan
- Ikuti peninjauan kode dan proses persetujuan organisasi Anda

Akses repositori kode

AWS Security Agent dapat memberikan panduan keamanan tentang perubahan kode melalui komentar permintaan tarik dan integrasi tinjauan kode. Untuk melindungi informasi keamanan sensitif, fungsi ini beroperasi di bawah batasan tertentu.

Batasan: Panduan keamanan kode dibatasi hanya untuk repositori pribadi. Ini memastikan bahwa:

- Temuan keamanan tetap rahasia bagi organisasi Anda
- Potensi kerentanan tidak diungkapkan secara publik sebelum remediasi
- Rekomendasi perbaikan yang dihasilkan tidak mengekspos detail eksploitasi

AWS Security Agent tidak memberikan panduan keamanan kode untuk repositori publik. Ini tidak akan mengomentari repositori publik atau proyek sumber terbuka di mana temuan keamanan akan terlihat oleh publik.

Tes penetrasi AWS Security Agent dapat meninjau dan memulihkan repositori pribadi dan publik yang Anda konfigurasikan untuk pentest. Jika repositori bersifat publik, kode remediasi akan disediakan sebagai file diff yang dapat diunduh alih-alih permintaan tarik.

URL yang Dapat Diakses

URL yang dapat diakses menentukan titik akhir tambahan yang dapat diakses oleh lingkungan pengujian penetrasi selama pengujian. Ini diperlukan ketika aplikasi Anda bergantung pada layanan eksternal seperti penyedia otentikasi pihak ketiga atau CDN. Semua dependensi jaringan yang diperlukan untuk pengujian harus ditentukan sebagai URL target atau URL yang dapat diakses. Jaringan memblokir akses ke titik akhir yang tidak ditentukan.

Implikasi keamanan: AWS Security Agent tidak diinstruksikan untuk melakukan pengujian keamanan pada URL yang dapat diakses. Dengan menentukan URL yang dapat diakses, Anda menunjukkan kepercayaan pada dependensi ini. Data uji penetrasi, termasuk kredensyal, dapat dikirimkan ke titik akhir URL yang dapat diakses ini selama pengujian.

Inferensi Lintas Wilayah

AWS Security Agent secara otomatis memilih Wilayah optimal untuk memproses permintaan inferensi Anda. Ini memaksimalkan sumber daya komputasi yang tersedia, ketersediaan model, dan memberikan pengalaman pelanggan terbaik. Data Anda tetap disimpan hanya di Wilayah tempat permintaan berasal; namun, permintaan input dan hasil keluaran mungkin diproses di luar Wilayah tersebut. Kami mengirimkan semua data yang dienkripsi di seluruh jaringan AWS.

AWS Security Agent menggunakan dua jenis inferensi lintas wilayah tergantung pada Wilayah:

- Inferensi lintas wilayah geografis - Menjaga pemrosesan data dalam batas geografis tertentu (seperti AS, UE, Australia, atau Jepang) untuk sebagian besar fitur. Untuk Code Remediation, permintaan dari Australia dan Jepang diproses di Uni Eropa. Digunakan di AS Timur (Virginia N.) —`us-east-1`, AS Barat (Oregon) —`us-west-2`, Asia Pasifik (Sydney) —`ap-southeast-2`, Asia Pasifik (Tokyo) —`ap-northeast-1`, Eropa (Frankfurt) —`eu-central-1`, dan Eropa (Irlandia) —. `eu-west-1`
- Inferensi lintas wilayah global — Merutekan permintaan inferensi ke Wilayah AWS komersial mana pun, mengoptimalkan sumber daya yang tersedia, dan memungkinkan throughput model yang lebih tinggi. Digunakan di Asia Pasifik (Mumbai) —`ap-south-1`, Asia Pasifik (Singapura) —`ap-southeast-1`, dan Amerika Selatan (São Paulo) —. `sa-east-1`

Untuk Wilayah yang menggunakan inferensi lintas wilayah global, permintaan input dan hasil keluaran dapat diproses di Wilayah AWS komersial mana pun. Semua data yang dikirimkan selama operasi lintas wilayah tetap berada di jaringan AWS dan tidak melintasi internet publik. Kami mengenkripsi data dalam perjalanan antar Wilayah AWS.

Tabel berikut menjelaskan di mana permintaan inferensi Anda diproses berdasarkan Wilayah tempat permintaan tersebut berasal dan fitur yang digunakan.

Minta asal	Semua fitur kecuali Remediasi Kode	Remediasi Kode
Amerika Serikat - AS Timur (Virginia N.) `us-east-1` -, AS Barat (Oregon) - `us-west-2`	Amerika Serikat	Amerika Serikat
Uni Eropa - Eropa (Irlandia) `eu-west-1` -, Eropa (Frankfurt) - `eu-central-1`	Uni Eropa	Uni Eropa
Australia - Asia Pasifik (Sydney) - `ap-southeast-2`	Australia	Uni Eropa
Jepang - Asia Pasifik (Tokyo) - `ap-northeast-1`	Jepang	Uni Eropa
Amerika Selatan - Amerika Selatan (São Paulo) - `sa-east-1`	Wilayah AWS komersial apa pun	Wilayah AWS komersial apa pun
India - Asia Pasifik (Mumbai) - `ap-south-1`	Wilayah AWS komersial apa pun	Wilayah AWS komersial apa pun
Asia Tenggara - Asia Pasifik (Singapura) - `ap-southeast-1`	Wilayah AWS komersial apa pun	Wilayah AWS komersial apa pun

Cross-Region inferensi selalu diaktifkan dan tidak dapat dipilih. Cross-Region inferensi tidak terpengaruh oleh kebijakan pelanggan dalam Kebijakan Kontrol Layanan (SCP) atau AWS Control Tower yang membatasi konten pelanggan ke Wilayah tertentu. Untuk informasi selengkapnya tentang cara AWS Security Agent melindungi data Anda selama pemrosesan lintas wilayah, lihat pemrosesan Cross-Region data.

Tindakan IAM dan migrasi sumber daya

AWS Security Agent adalah agen perbatasan yang secara proaktif mengamankan aplikasi Anda selama siklus hidup pengembangan di semua lingkungan Anda. Jika Anda bergabung dengan AWS Security Agent sebelum 9 Februari 2026, Anda akan terpengaruh oleh perubahan yang akan datang pada 9 Maret 2026 pada sumber daya Instans Agen yang ada dan tindakan IAM Agen Keamanan AWS. Dalam persiapan untuk merilis API/SDK dukungan publik, sumber daya Instans Agen sedang diubah namanya menjadi Agent Space, dan tindakan IAM tertentu sedang diganti namanya. Perubahan ini akan memengaruhi peran IAM Aplikasi atau Instans Agen apa pun yang telah Anda buat sebelum 9 Maret 2026. Untuk menghindari melihat masalah otentikasi setelah 9 Maret 2026, Anda harus mengikuti langkah-langkah di bawah Mempersiapkan Migrasi.

> **Note**
>
> Jika Anda membuat Instans Agen baru setelah 9 Februari 2026, Instans Agen baru akan dibuat sebagai Ruang Agen dan tidak ada langkah migrasi yang diperlukan.

Perubahan yang Direncanakan

AWS Security Agent mengganti nama sumber daya Instans Agen menjadi Agent Space: `arn:aws:securityagent:us-east-1:{{accountId}}:agent-instance/*` diganti namanya menjadi. `arn:aws:securityagent:us-east-1:{{accountId}}:agent-space/*` Selain itu, tindakan IAM berikut sedang diganti namanya:

- `securityagent:ListAgentInstances`berganti nama menjadi `securityagent:ListAgentSpaces`
- `securityagent:ListControls`berganti nama menjadi `securityagent:ListSecurityRequirements`
- `securityagent:BatchGetAgentInstances`berganti nama menjadi `securityagent:BatchGetAgentSpaces`
- `securityagent:BatchGetSecurityTestContentMetadata`berganti nama menjadi `securityagent:BatchGetPentestJobContentMetadata`
- `securityagent:BatchGetTasks`berganti nama menjadi `securityagent:BatchGetPentestJobTasks`
- `securityagent:CreateDocumentReview`berganti nama menjadi `securityagent:CreateDesignReview`

- `securityagent:GetDocumentReview`berganti nama menjadi `securityagent:GetDesignReview`
- `securityagent:GetDocumentReviewArtifact`berganti nama menjadi `securityagent:GetDesignReviewArtifact`
- `securityagent:ListDocumentReviews`berganti nama menjadi `securityagent:ListDesignReviews`
- `securityagent:ListDocumentReviewComments`berganti nama menjadi `securityagent:ListDesignReviewComments`
- `securityagent:ListTasks`berganti nama menjadi `securityagent:ListPentestJobTasks`
- `securityagent:StartPentestExecution`berganti nama menjadi `securityagent:StartPentestJob`
- `securityagent:StopPentestExecution`berganti nama menjadi `securityagent:StopPentestJob`
- `securityagent:DeleteDocumentReview`berganti nama menjadi `securityagent:DeleteDesignReview`

Mempersiapkan Migrasi

Untuk menghindari masalah setelah 9 Maret 2026 sambil terus menggunakan AWS Security Agent sebelum 9 Maret 2026, Anda harus mempercayai sumber daya baru dan lama serta tindakan IAM di IAM Anda roles/policies hingga 9 Maret 2026. Petunjuk di bawah ini akan memberikan panduan untuk bermigrasi ke format sumber daya dan tindakan baru:

1. Masuk ke akun AWS Anda dan navigasikan ke konsol AWS Security Agent
2. Di panel sebelah kiri, pilih Pengaturan dan pilih peran di bawah Peran layanan
3. Di konsol IAM untuk peran terkait, pilih Tambahkan izin dan Lampirkan kebijakan
4. Pilih AWSSecurityAgentWebAppPolicydan pilih Tambahkan izin
   a. Catatan Penting: Verifikasi bahwa Anda telah memilih AWSSecurityAgentWebAppPolicysebagai kebijakan baru dan bukan SecurityAgentWebAppAPIPolicy
5. Verifikasi bahwa peran IAM Anda sekarang memiliki keduanya AWSSecurityAgentWebAppPolicydan SecurityAgentWebAppAPIPolicydi bawah kebijakan Izin
6. Di konsol peran IAM yang sama, pilih Hubungan kepercayaan lalu Edit kebijakan kepercayaan
7. Perbarui kebijakan kepercayaan Anda ke format berikut, ganti {{accountID}} dengan ID akun AWS

```
{
  "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. Arahkan kembali ke konsol AWS Security Agent. Dari panel sebelah kiri, pilih Agent Spaces
9. Untuk setiap Ruang Agen yang Anda miliki dengan pengujian penetrasi diaktifkan, lakukan langkah-langkah berikut
   a. Arahkan ke Ruang Agen dan pilih Tes Penetrasi
   b. Di bawah Akses layanan, pilih peran di bawah Nama peran Layanan
   c. Di konsol IAM untuk peran terkait, pilih Hubungan kepercayaan lalu Edit kebijakan kepercayaan
   d. Perbarui kebijakan kepercayaan Anda ke format berikut, ganti {{accountID}} dengan ID akun AWS

```
{
  "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/*"
          ]
        }
      }
    }
  ]
}
```

Pencatatan log AWS Panggilan API Agen Keamanan dengan AWS CloudTrail

AWS Agen Keamanan terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Agen AWS Keamanan. CloudTrail menangkap semua panggilan API untuk Agen AWS Keamanan sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Agen AWS Keamanan dan panggilan kode ke operasi API Agen AWS Keamanan. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk Agen AWS Keamanan. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Agen AWS Keamanan, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS Informasi Agen Keamanan di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Agen AWS Keamanan, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Agen AWS Keamanan, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan Agen AWS Keamanan dicatat oleh CloudTrail. Misalnya, panggilan ke`CreatePentest`,`BatchGetPentestJobs`, dan `ListFindings` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan dibuat dengan root atau kredensyal pengguna AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Contoh: AWS Entri file log Agen Keamanan

Memahami AWS Entri file log Agen Keamanan

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreatePentest` tindakan:

```
{
    "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 Agent memiliki kuota yang membatasi jumlah sumber daya yang dapat Anda buat dan tingkat di mana Anda dapat melakukan operasi. Kuota yang ditandai sebagai dapat disesuaikan dapat ditingkatkan dengan mengirimkan permintaan melalui AWS Support. Untuk informasi selengkapnya, lihat Membuat kasus dukungan dan manajemen kasus.

Kuota Operasi

Kuota operasi membatasi penggunaan bulanan pengujian keamanan dan fitur peninjauan untuk membantu mengelola kapasitas layanan.

Sumber daya	Lingkup	Kuota	Dapat Disesuaikan
Ulasan desain	Per bulan per akun per wilayah	200	Ya
Ulasan kode PR	Per bulan per akun per wilayah	1.000	Ya

Kuota Konfigurasi

Kuota konfigurasi membatasi jumlah sumber daya dan pengaturan yang dapat Anda konfigurasikan di lingkungan AWS Security Agent Anda.

Sumber daya	Lingkup	Kuota	Dapat Disesuaikan
Agen Spaces	Per akun per wilayah	100	Ya
Integrasi	Per akun per wilayah	20	Tidak
Sumber daya terintegrasi per integrasi	Per integrasi	50	Tidak
Persyaratan keamanan khusus	Per akun per wilayah	20	Tidak
Proyek pentest	Per akun per wilayah	1.000	Ya
Pentest berjalan bersamaan	Per akun per wilayah	5	Ya
Proyek peninjauan kode	Per akun per wilayah	1.000	Ya
Tinjauan kode bersamaan berjalan	Per akun per wilayah	5	Ya

Riwayat dokumen

Tabel berikut menjelaskan beberapa pembaruan utama dan fitur baru untuk Panduan Pengguna Agen Keamanan AWS.

Perubahan	Deskripsi	Tanggal
Koneksi pribadi (pratinjau)	Ditambahkan dokumentasi untuk koneksi pribadi. Kemampuan baru ini memungkinkan AWS Security Agent untuk terhubung ke sistem kontrol sumber yang berjalan di jaringan pribadi menggunakan Amazon VPC Lattice, tanpa mengekspos sistem Anda ke internet publik.	Juni 16, 2026
Pemodelan ancaman (pratinjau)	Menambahkan dokumentasi untuk pemodelan ancaman. Kemampuan baru ini membangun model ancaman aplikasi Anda dari dokumen desain (dokumen lingkup), kode sumber (sumber), atau keduanya. Dokumen lingkup adalah dokumen desain fitur yang menentukan fokus analisis, sementara kode sumber menyediakan konteks tentang sistem Anda yang ada. Jika Anda tidak menyediakan dokumen cakupan, agen menghasilkan model ancaman dari kode sumber saja. Setiap proses menghasilkan ikhtisar	Juni 8, 2026

	sistem dan serangkaian ancaman yang diklasifi kasikan berdasarkan kategori STRIDE dengan peringkat dan rekomendasi tingkat keparahan.	
Ulasan kode repositori lengkap (pratinjau)	Ditambahkan dokumentasi untuk review kode repositor i lengkap. Kemampuan baru ini melakukan analisis keamanan sadar konteks dari seluruh basis kode Anda dan menghasilkan remediasi kode untuk temuan.	12 Mei 2026
Ketersediaan Wilayah yang diperbarui untuk menemukan remediasi	Ketersediaan wilayah untuk remediasi pencarian pengujian penetrasi di Aplikasi Web AWS Security Agent telah diperbarui.	April 16, 2026
Ketersediaan Wilayah yang diperbarui untuk memungkin kan perbaikan pencarian	Ketersediaan wilayah untuk mengaktifkan remediasi pencarian pengujian penetrasi di AWS Management Console telah diperbarui.	April 16, 2026

Dukungan kunci yang dikelola pelanggan	Menambahkan dokumentasi untuk dukungan kunci terkelola pelanggan (CMK). Anda sekarang dapat menentukan kunci KMS yang dikelola pelanggan saat membuat Ruang Agen dan integrasi untuk mengenkripsi data Anda dengan kunci yang Anda kontrol.	Maret 31, 2026
AWS pembaruan kebijakan terkelola	Menambahkan kebijakan AWSSecurityAgentWebAppPolicy terkelola untuk tipe baru TargetDomain dan DesignReviewFeedback sumber daya.	Maret 31, 2026
AWS pembaruan kebijakan terkelola	Menambahkan kebijakan AWSSecurityAgentWebAppPolicy terkelola untuk jenis AgentSpace sumber daya baru dan perubahan nama tindakan IAM.	Februari 9, 2026
AWS pembaruan kebijakan terkelola	Diperbarui SecurityAgentWebAppAPIPolicy untuk memungkinkan pelanggan menghapus ulasan desain.	Januari 28, 2026
AWS pembaruan kebijakan terkelola	Diperbarui SecurityAgentWebAppAPIPolicy untuk memungkinkan pelanggan memulai remediasi kode otomatis untuk temuan keamanan.	Januari 20, 2026

AWS pembaruan kebijakan terkelola	Diperbarui SecurityAgentWebAppAPIPolicy untuk memungkinkan pelanggan melihat gambar di konsol.	Desember 5, 2025
AWS Security Agent rilis awal	Dokumentasi awal untuk peluncuran layanan	Desember 2, 2025

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.