AWS Well-Architected Framework

Pilar Keamanan

Pilar Keamanan: AWS Well-Architected Framework

Table of Contents

Pilar Keamanan - Kerangka Kerja AWS Well-Architected

Tanggal publikasi: 6 Desember 2023 (Revisi Dokumen)

Laporan ini berfokus pada pilar keamanan AWS Well-Architected Framework. Laporan ini menyediakan panduan untuk membantu Anda menerapkan praktik terbaik, rekomendasi terkini dalam hal desain, penyediaan, dan pemeliharaan beban kerja AWS.

Pengantar

Dengan AWS Well-Architected Framework Anda dapat memahami kompromi untuk keputusan yang Anda ambil selama membangun beban kerja di AWS. Dengan menggunakan Kerangka Kerja ini, Anda akan mengetahui praktik terbaik arsitektur terkini untuk mendesain dan mengoperasikan beban kerja yang andal, aman, efisien, hemat biaya, dan ramah lingkungan di cloud. Kerangka kerja ini menyediakan cara untuk secara terus menerus menilai beban kerja Anda berdasarkan praktik terbaik dan mengidentifikasi area yang perlu diperbaiki. Kami percaya bahwa memiliki beban kerja yang didesain dengan baik akan meningkatkan peluang keberhasilan bisnis.

Enam pilar landasan kerangka kerja:

- Keunggulan Operasional
- Keamanan
- Keandalan
- Efisiensi Kinerja
- Optimasi Biaya
- Pelestarian Lingkungan

Laporan ini berfokus pada pilar keamanan. Ini akan membantu Anda memenuhi persyaratan bisnis dan peraturan dengan mengikuti saran AWS terkini. Dokumen ini dimaksudkan untuk orang-orang yang memiliki peran di bidang teknologi, seperti kepala pejabat teknologi (CTO), kepala pejabat keamanan informasi (CSO/CISO), arsitek, pengembang, dan anggota tim operasi.

Setelah membaca laporan ini, Anda akan memahami saran dan strategi AWS terkini untuk digunakan ketika merancang arsitektur cloud dengan mempertimbangkan keamanan. Laporan ini tidak menyediakan detail implementasi atau pola arsitektural, tetapi menyertakan referensi ke sumber daya yang relevan untuk informasi ini. Dengan mengadopsi praktik-praktik dalam laporan ini, Anda dapat

membangun arsitektur yang melindungi data dan sistem Anda, mengontrol akses, dan merespons peristiwa keamanan secara otomatis.

Fondasi keamanan

Pilar keamanan menjelaskan cara memanfaatkan teknologi cloud untuk melindungi data, sistem, dan aset guna meningkatkan postur keamanan Anda. Dokumen ini menyediakan panduan praktik terbaik yang mendalam tentang perancangan beban kerja yang aman di AWS.

Prinsip desain

Ada sejumlah prinsip di cloud yang dapat membantu Anda memperkuat keamanan beban kerja Anda:

- Menerapkan landasan identitas yang kuat: Implementasikan prinsip hak akses paling rendah dan berlakukan pemisahan tugas dengan otorisasi yang sesuai untuk setiap interaksi dengan sumber daya AWS Anda. Pusatkan manajemen identitas, dan targetkan untuk tidak bergantung pada kredensial statis jangka panjang.
- Pertahankan keterlacakan: Pantau, munculkan peringatan, dan audit tindakan serta perubahan dalam lingkungan Anda secara waktu nyata. Integrasikan pengumpulan log dan metrik dengan sistem agar dapat bertindak berdasarkan investigasi yang berjalan otomatis.
- Menerapkan keamanan di semua lapisan: Terapkan pertahanan secara mendalam dengan banyak kontrol keamanan. Terapkan ke semua lapisan (misalnya, edge jaringan, VPC, penyeimbangan beban, setiap layanan komputasi dan instans, sistem operasi, aplikasi, dan kode).
- Mengotomatiskan praktik terbaik keamanan: Mekanisme keamanan berbasis perangkat lunak otomatis meningkatkan kemampuan Anda untuk meningkatkan skala dengan lebih cepat, hemat biaya, dan aman. Ciptakan arsitektur yang aman, termasuk implementasi kontrol yang ditentukan dan dikelola sebagai kode dalam templat yang dikontrol versi.
- Melindungi data bergerak dan data diam: Klasifikasikan data sesuai tingkatan sensitivitasnya dan gunakan mekanisme, seperti enkripsi, tokenisasi, dan kontrol akses jika sesuai.
- Minimalkan campur tangan manusia dari data: Gunakan mekanisme dan alat untuk mengurangi atau meniadakan akses langsung atau pemrosesan data secara manual. Ini akan mengurangi risiko kekeliruan atau perubahan dan kesalahan manusia dalam penanganan data sensitif.
- Bersiap untuk peristiwa keamanan: Bersiaplah menghadapi insiden dengan membentuk manajemen insiden serta proses dan kebijakan investigasi yang selaras dengan kebutuhan organisasi Anda. Jalankan simulasi tanggap-insiden dan gunakan alat dengan otomatisasi untuk mempercepat deteksi, investigasi, dan pemulihan.

Definisi

Keamanan di cloud terdiri dari tujuh area:

- Fondasi keamanan
- Manajemen identitas dan akses
- Deteksi
- Perlindungan infrastruktur
- Perlindungan data
- Respons insiden
- Keamanan aplikasi

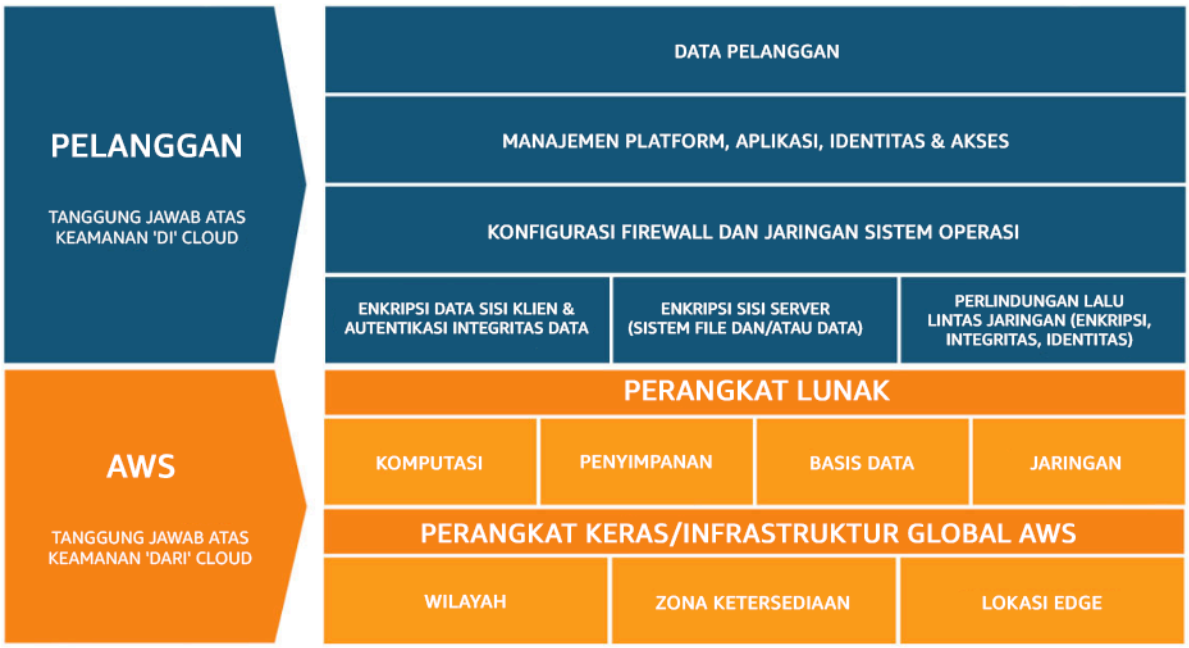
Tanggung jawab bersama

Keamanan dan kepatuhan merupakan tanggung jawab bersama antara AWS dan pelanggan. Model bersama seperti ini dapat membantu meringankan beban operasional pelanggan karena AWS mengoperasikan, mengelola, dan mengendalikan komponen dari sistem operasi dan lapisan virtualisasi host hingga keamanan fisik dari fasilitas tempat layanan tersebut beroperasi. Pelanggan meneruskan tanggung jawab dan manajemen pada sistem operasi tamu (termasuk pembaruan dan patch keamanan), aplikasi perangkat lunak terkait lainnya, dan juga konfigurasi firewall grup keamanan yang disediakan oleh AWS. Pelanggan harus dengan cermat mempertimbangkan layanan yang mereka pilih karena tanggung jawab mereka sangat bergantung pada layanan yang digunakan, integrasi dari layanan tersebut ke dalam lingkungan IT mereka, serta undang-undang dan regulasi yang berlaku. Pada dasarnya, tanggung jawab bersama ini juga menyediakan kontrol pelanggan dan fleksibilitas yang mengizinkan deployment. Sebagaimana ditunjukkan pada bagan berikut, pembedaan tanggung jawab ini umumnya disebut sebagai Keamanan “dari” Cloud versus Keamanan “dalam” Cloud.

Tanggung jawab AWS untuk “Keamanan dari Cloud” – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan di AWS Cloud. Infrastruktur ini terdiri dari perangkat keras, perangkat lunak, jaringan, dan fasilitas yang menjalankan layanan AWS Cloud.

Tanggung jawab pelanggan untuk “Keamanan di dalam Cloud” – Tanggung jawab pelanggan akan ditentukan oleh layanan AWS Cloud yang dipilih pelanggan. Hal ini menentukan jumlah tugas konfigurasi yang harus dilakukan pelanggan sebagai bagian dari tanggung jawab keamanan

mereka. Misalnya, layanan seperti Amazon Elastic Compute Cloud (Amazon EC2) dikategorikan sebagai Infrastruktur sebagai Layanan (IaaS) dan, oleh karena itu, pelanggan wajib melakukan semua konfigurasi keamanan dan tugas manajemen yang diperlukan. Jika pelanggan melakukan deployment instans Amazon EC2, mereka bertanggung jawab atas manajemen sistem operasi tamu (termasuk pembaruan dan patch keamanan), semua aplikasi perangkat lunak atau utilitas yang diinstal pelanggan pada instans, dan konfigurasi firewall yang disediakan AWS (yang disebut grup keamanan) pada setiap instans. Untuk layanan yang diabstraksi, seperti Amazon S3 dan Amazon DynamoDB, AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, sedangkan pelanggan mengakses titik akhir untuk menyimpan dan mengambil data. Pelanggan bertanggung jawab mengelola data mereka (termasuk opsi enkripsi), mengklasifikasikan aset mereka, dan menggunakan alat IAM untuk menerapkan izin yang sesuai.



Gambar 1: Model Tanggung Jawab Bersama AWS.

Model tanggung jawab bersama antara pelanggan/AWS ini juga mencakup kontrol IT. Selain dalam mengoperasikan lingkungan IT, tanggung jawab bersama antara AWS dan pelanggannya pun juga mencakup manajemen, operasi, dan verifikasi kontrol IT bersama. AWS dapat meringankan beban pelanggan dalam mengoperasikan kontrol dengan cara mengelola kontrol yang terkait dengan infrastruktur fisik yang di-deploy di lingkungan AWS yang mungkin sebelumnya dikelola oleh pelanggan. Karena setiap pelanggan melakukan deployment dengan cara yang berbeda-beda di AWS, pelanggan dapat mengalihkan manajemen untuk kontrol IT tertentu ke AWS, sehingga menciptakan lingkungan kontrol terdistribusi (yang baru). Pelanggan kemudian dapat menggunakan dokumentasi kontrol dan kepatuhan AWS untuk melakukan evaluasi kontrol dan prosedur verifikasi

sendiri sebagaimana diperlukan. Berikut ini adalah contoh kontrol yang dikelola oleh AWS, pelanggan AWS, atau keduanya.

Kontrol Warisan – Kontrol yang diwariskan sepenuhnya oleh AWS kepada pelanggan.

- Kontrol Fisik dan Lingkungan

Kontrol Bersama – Kontrol yang diterapkan ke lapisan infrastruktur dan lapisan pelanggan, tetapi dalam konteks atau perspektif terpisah. Dalam kontrol bersama, AWS menyediakan persyaratan untuk infrastruktur dan pelanggan harus menyediakan implementasi kontrolnya sendiri dalam penggunaan mereka atas layanan AWS. Contohnya mencakup:

- Manajemen Patch – AWS bertanggung jawab melakukan patching dan memperbaiki kelemahan dalam infrastruktur, tetapi pelanggan bertanggung jawab melakukan patching untuk aplikasi dan sistem operasi tamu mereka.
- Manajemen Konfigurasi – AWS mengurus konfigurasi perangkat infrastrukturnya, tetapi pelanggan bertanggung jawab mengonfigurasi basis data, aplikasi, dan sistem operasi tamu mereka.
- Kesadaran dan Pelatihan – AWS melatih karyawan AWS, tetapi pelanggan harus melatih karyawan mereka sendiri.

Spesifik untuk Pelanggan – Kontrol yang sepenuhnya merupakan tanggung jawab pelanggan berdasarkan aplikasi yang mereka deploy dalam layanan AWS. Contohnya mencakup:

- Perlindungan Layanan dan Komunikasi atau Keamanan Zona, yang mungkin mewajibkan pengguna untuk merutekan atau membuat zona data dalam lingkungan keamanan tertentu.

Tata kelola

Tata kelola keamanan, sebagai subset pendekatan keseluruhan, dimaksudkan untuk mendukung tujuan bisnis dengan menentukan tujuan kebijakan dan kontrol untuk membantu mengelola risiko. Bentuk manajemen risiko dengan mengikuti pendekatan berlapis terhadap tujuan kontrol keamanan–setiap lapisan menutupi lapisan di bawahnya. Memahami bahwa Model Tanggung Jawab Bersama AWS adalah lapisan fondasi Anda. Pemahaman ini memberikan kejelasan atas apa yang menjadi tanggung jawab Anda dari sisi pelanggan dan apa yang Anda warisi dari AWS. Satu sumber daya yang berguna adalah AWS Artifact, yang memberikan akses sesuai permintaan ke laporan keamanan dan kepatuhan AWS serta untuk memilih perjanjian online.

Penuhi sebagian besar tujuan kontrol Anda di lapisan berikutnya. Di lapisan inilah kemampuan tingkat platform berada. Misalnya, lapisan ini mencakup proses vending akun AWS, integrasi dengan penyedia identitas seperti AWS IAM Identity Center, dan kontrol deteksi umum. Beberapa output dari proses tata kelola platform juga ada di sini. Ketika Anda ingin mulai menggunakan layanan AWS baru, perbarui kebijakan kontrol layanan (SCP) di layanan AWS Organizations guna menyediakan pagar pembatas untuk penggunaan awal layanan tersebut. Anda dapat menggunakan SCP lainnya untuk menerapkan sasaran kontrol keamanan umum lainnya, yang sering kali disebut sebagai invarian keamanan. Ini adalah sasaran atau konfigurasi kontrol yang Anda terapkan ke banyak akun, unit organisasi, atau keseluruhan organisasi AWS. Contoh umumnya adalah membatasi Wilayah tempat infrastruktur berjalan atau mencegah penonaktifkan kontrol deteksi. Lapisan tengah ini juga berisi kebijakan terkodifikasi seperti aturan konfigurasi atau alur pemeriksaan masuk.

Lapisan teratas adalah tempat tim produk memenuhi sasaran kontrol. Ini karena implementasi dilakukan di aplikasi yang dikontrol oleh tim produk. Ini bisa berupa implementasi validasi input dalam aplikasi atau memastikan bahwa identitas diteruskan dengan benar antarlayanan mikro. Meskipun konfigurasi dimiliki oleh tim produk, mereka tetap dapat mewarisi beberapa kemampuan dari lapisan tengah.

Di mana pun Anda mengimplementasikan kontrol, tujuannya sama: mengelola risiko. Serangkaian kerangka kerja manajemen risiko berlaku pada industri, wilayah, atau teknologi tertentu. Tujuan utama Anda: menyoroti risiko berdasarkan kemungkinan dan konsekuensi. Ini adalah risiko inheren. Anda kemudian dapat menentukan tujuan kontrol yang mengurangi kemungkinan, konsekuensi, atau keduanya. Lalu, ketika kontrol sudah ada, Anda dapat melihat seperti apa kecenderungan risikonya. Ini adalah risiko residual. Tujuan kontrol dapat berlaku pada satu atau banyak beban kerja. Diagram berikut ini menampilkan matriks risiko tipikal. Kecenderungannya didasarkan pada frekuensi kejadian sebelumnya dan konsekuensinya didasarkan pada kerugian keuangan, reputasi, dan waktu atas peristiwa tersebut.

Tingkat Kemungkinan	Tingkat Risiko				
Sangat besar kemungkinan	Rendah	Sedang	Tinggi	Kritis	Kritis
Besar Kemungkinan	Rendah	Sedang	Sedang	Tinggi	Kritis
Mungkin	Rendah	Rendah	Sedang	Sedang	Tinggi
Kecil kemungkinan	Rendah	Rendah	Sedang	Sedang	Tinggi
Sangat kecil kemungkinan	Rendah	Rendah		Sedang	Tinggi
Konsekuensi	Minimal	Rendah	Sedang	Tinggi	Sangat berat

Gambar 2: Matriks kecenderungan tingkat risiko

Manajemen dan pemisahan akun AWS

Sebaiknya atur beban kerja di akun dan akun grup terpisah berdasarkan fungsi, persyaratan kepatuhan, atau serangkaian kontrol umum, daripada menyamakannya dengan struktur pelaporan perusahaan Anda. Di AWS, akun adalah batas tegas. Misalnya, pemisahan di tingkat akun sangat disarankan untuk mengisolasi beban kerja produksi dari beban kerja pengembangan dan pengujian.

Mengelola akun secara terpusat: AWS Organizations mengotomatiskan pembuatan dan pengelolaan akun AWS, serta kontrol terhadap akun tersebut setelah dibuat. Ketika Anda membuat akun melalui AWS Organizations, pertimbangkan dengan cermat alamat email yang Anda gunakan, karena ini akan menjadi pengguna root yang dapat mengatur ulang kata sandi. Organizations memungkinkan Anda mengelompokkan akun ke dalam unit organisasi (OU), yang dapat merepresentasikan berbagai lingkungan berdasarkan persyaratan dan tujuan beban kerja.

Menetapkan kontrol secara terpusat: Kontrol apa saja yang dapat dilakukan akun AWS Anda dengan hanya memperbolehkan layanan, Wilayah, dan tindakan layanan tertentu di tingkat yang sesuai. AWS Organizations memungkinkan Anda menggunakan kebijakan kontrol layanan (SCP) untuk menerapkan pagar pembatas di tingkat organisasi, unit organisasi, atau akun, yang berlaku

untuk semua pengguna dan peran AWS Identity and Access Management (IAM). Misalnya, Anda dapat menerapkan SCP yang membatasi pengguna agar tidak dapat meluncurkan sumber daya di Wilayah yang tidak Anda izinkan secara eksplisit. AWS Control Tower menawarkan cara sederhana untuk menyiapkan dan mengatur banyak akun. Hal ini mengotomatiskan penyiapan akun di AWS Organization Anda, mengotomatiskan penyediaan, menerapkan pagar pembatas (mencakup pencegahan dan deteksi), dan menyediakan dasbor untuk memudahkan Anda.

Mengonfigurasi layanan dan sumber daya secara terpusat: AWS Organizations membantu Anda mengonfigurasi layanan AWS yang berlaku ke semua akun Anda. Misalnya, Anda dapat mengonfigurasi log terpusat untuk semua tindakan yang dilakukan di organisasi Anda menggunakan AWS CloudTrail, dan mencegah akun anggota menonaktifkan pencatatan log. Anda juga dapat mengagregasi data secara terpusat untuk aturan yang Anda tetapkan menggunakan AWS Config, yang memungkinkan Anda untuk mengaudit beban kerja untuk kepatuhan dan menanggapi perubahan dengan cepat. AWS CloudFormation StackSets memudahkan Anda untuk mengelola tumpukan AWS CloudFormation secara terpusat di berbagai akun dan unik organisasi dalam organisasi Anda. Dengan begitu, Anda dapat secara otomatis menyediakan akun baru yang memenuhi persyaratan keamanan Anda.

Gunakan fitur administrasi terdelegasikan dari layanan keamanan untuk memisahkan akun yang digunakan untuk manajemen dari akun tagihan (manajemen) organisasi. Beberapa layanan AWS, seperti GuardDuty, Security Hub, dan AWS Config, mendukung integrasi dengan AWS Organizations, termasuk menentukan akun spesifik untuk fungsi administratif.

Praktik terbaik

- SEC01-BP01 Memisahkan beban kerja menggunakan akun
- SEC01-BP02 Mengamankan properti dan pengguna root akun

SEC01-BP01 Memisahkan beban kerja menggunakan akun

Terapkan pagar pembatas umum dan isolasi antarlingkungan (seperti produksi, pengembangan, dan pengujian) dan beban kerja melalui strategi multiakun. Pemisahan di tingkat akun sangat disarankan karena hal ini dapat memberikan batasan isolasi yang kuat untuk keamanan, tagihan, dan akses.

Hasil yang diinginkan: Struktur akun yang mengisolasi operasi cloud, beban kerja yang tidak saling berkaitan, dan lingkungan dalam akun terpisah, sehingga meningkatkan keamanan di seluruh infrastruktur cloud.

Antipola umum:

- Menempatkan beberapa beban kerja yang tidak saling berkaitan dengan berbagai tingkat sensitivitas data ke dalam akun yang sama.
- Struktur unit organisasi (OU) yang tidak ditentukan dengan baik.

Manfaat menjalankan praktik terbaik ini:

- Mengurangi cakupan dampak jika beban kerja tidak sengaja diakses.
- Tata kelola akses secara terpusat ke layanan, sumber daya, dan Wilayah AWS.
- Keamanan infrastruktur cloud terjaga dengan kebijakan dan administrasi terpusat pada layanan keamanan.
- Pembuatan akun dan proses pemeliharaan otomatis.
- Audit infrastruktur terpusat untuk persyaratan kepatuhan dan peraturan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Akun AWS memberikan batas isolasi keamanan di antara beban kerja atau sumber daya yang beroperasi pada tingkat sensitivitas yang berbeda. AWS menyediakan alat untuk mengelola beban kerja cloud Anda dalam skala besar melalui strategi multiakun untuk memanfaatkan batasan isolasi ini. Untuk panduan tentang konsep, pola, dan implementasi strategi multiakun di AWS, lihat Mengelola Lingkungan AWS Anda Menggunakan Beberapa Akun.

Ketika Anda memiliki beberapa Akun AWS di bawah manajemen pusat, akun Anda harus dikelola dalam hierarki yang ditentukan oleh lapisan unit organisasi (OU). Kontrol keamanan kemudian dapat diatur dan diterapkan ke OU dan akun anggota, yang menciptakan kontrol pencegahan yang konsisten pada akun anggota di organisasi. Kontrol keamanan diwariskan, memungkinkan Anda memfilter izin yang tersedia untuk akun anggota yang berada di tingkat yang lebih rendah dalam hierarki OU. Untuk membuat desain yang baik, memanfaatkan pewarisan ini untuk mengurangi jumlah dan kerumitan kebijakan keamanan yang diperlukan untuk mencapai kontrol keamanan yang diinginkan untuk setiap akun anggota.

AWS Organizations dan AWS Control Tower adalah dua layanan yang dapat Anda gunakan untuk mengimplementasikan dan mengelola struktur multiakun ini di lingkungan AWS Anda. Dengan AWS Organizations, Anda bisa mengatur akun ke dalam hierarki yang ditentukan oleh satu atau beberapa lapisan OU, dan setiap OU berisi sejumlah akun anggota. Kebijakan kontrol layanan (SCP) memungkinkan administrator organisasi untuk menetapkan kontrol pencegahan terperinci pada

akun anggota, dan AWS Config dapat digunakan untuk membuat kontrol proaktif dan detektif pada akun anggota. Banyak layanan AWS berintegrasi dengan AWS Organizations untuk memberikan kontrol administratif terdelegasi dan melakukan tugas khusus layanan di semua akun anggota dalam organisasi.

Selain AWS Organizations, AWS Control Tower menyediakan penyiapan praktik terbaik sekali klik untuk lingkungan AWS multiakun dengan zona landasan. Zona landasan adalah titik masuk ke lingkungan multiakun yang dibuat oleh Control Tower. Control Tower memiliki beberapa manfaat dibanding AWS Organizations. Tiga manfaat yang memberikan tata kelola akun yang lebih baik adalah:

- Pagar pembatas wajib terintegrasi yang diterapkan secara otomatis ke akun yang diterima di organisasi.
- Pagar pembatas opsional yang dapat diaktifkan atau dinonaktifkan untuk serangkaian OU tertentu.
- AWS Control Tower Account Factory menyediakan deployment otomatis untuk akun yang berisi dasar dan opsi konfigurasi yang telah disetujui sebelumnya di dalam organisasi Anda.

Langkah implementasi

1. Rancang struktur unit organisasi: Rancangan struktur organisasi yang tepat dapat mengurangi beban manajemen yang diperlukan untuk membuat dan mengelola kebijakan kontrol layanan dan kontrol keamanan lainnya. Struktur unit organisasi Anda harus selaras dengan struktur beban kerja, sensitivitas data, dan kebutuhan bisnis Anda.
2. Buat zona landasan untuk lingkungan multiakun: Zona landasan membentuk konsistensi keamanan dan landasan infrastruktur, sehingga organisasi Anda dapat dengan cepat mengembangkan, meluncurkan, dan melakukan deployment beban kerja. Anda dapat menggunakan zona landasan kustom atau AWS Control Tower untuk mengatur lingkungan Anda.
3. Terapkan pagar pembatas: Implementasikan pagar pembatas yang stabil untuk lingkungan Anda melalui zona landasan. AWS Control Tower menyediakan sederet kontrol wajib dan opsional yang dapat di-deploy. Deployment kontrol wajib dilakukan secara otomatis saat mengimplementasikan Control Tower. Lihat daftar kontrol yang sangat direkomendasikan dan opsional, kemudian implementasikan kontrol yang sesuai dengan kebutuhan Anda.
4. Batasi akses ke Wilayah yang baru ditambahkan: Untuk Wilayah AWS baru, sumber daya IAM seperti pengguna dan peran hanya disebarkan ke Wilayah yang Anda tentukan. Tindakan ini dapat dilakukan melalui konsol saat menggunakan Control Tower, atau dengan menyesuaikan kebijakan izin IAM di AWS Organizations.

5. Pertimbangkan AWS CloudFormation StackSets: StackSets membantu Anda saat melakukan deployment kebijakan, peran, dan grup IAM ke Wilayah dan Akun AWS yang berbeda dari templat yang disetujui.

Sumber daya

Praktik Terbaik Terkait:

- SEC02-BP04 Mengandalkan penyedia identitas terpusat

Dokumen terkait:

- AWS Control Tower
- Panduan Audit Keamanan AWS
- Praktik Terbaik IAM
- Menggunakan CloudFormation StackSets untuk menyediakan sumber daya di beberapa wilayah dan Akun AWS
- Pertanyaan Umum Organizations
- Terminologi dan konsep AWS Organizations
- Praktik Terbaik untuk Kebijakan Kontrol Layanan dalam Lingkungan Multiakun AWS Organizations
- Panduan Referensi Manajemen Akun AWS
- Mengatur Lingkungan AWS Anda Menggunakan Beberapa Akun

Video terkait:

- Aktifkan adopsi AWS dalam skala besar dengan otomatisasi dan tata kelola
- Praktik Terbaik Keamanan dengan Cara Well-Architected
- Membangun dan Mengatur Banyak Akun menggunakan AWS Control Tower
- Mengaktifkan Control Tower untuk Organisasi yang Ada

Lokakarya terkait:

- Control Tower Immersion Day (Hari Imersi Control Tower)

SEC01-BP02 Mengamankan properti dan pengguna root akun

Pengguna root adalah pengguna yang memiliki hak istimewa paling banyak dalam Akun AWS, dengan akses administratif penuh ke semua sumber daya di dalam akun, dan dalam beberapa kasus tidak dapat dibatasi oleh kebijakan keamanan. Menonaktifkan akses terprogram ke pengguna root, menerapkan kontrol yang sesuai untuk pengguna root, serta tidak menggunakan pengguna root secara rutin membantu mengurangi risiko tersebarnya kredensial root secara tidak sengaja dan penyusupan di lingkungan cloud.

Hasil yang diharapkan: Mengamankan pengguna root membantu mengurangi kemungkinan terjadinya kerusakan yang disengaja maupun tidak disengaja akibat penyalahgunaan kredensial pengguna root. Menerapkan kontrol detektif juga dapat memberikan peringatan kepada personel yang tepat saat ada tindakan dilakukan menggunakan pengguna root.

Antipola umum:

- Menggunakan pengguna root untuk tugas selain yang memerlukan kredensial pengguna root.
- Tidak menguji rencana darurat secara rutin untuk memverifikasi fungsi infrastruktur, proses, dan personel penting dalam keadaan darurat.
- Hanya mempertimbangkan alur masuk akun biasa dan tidak mempertimbangkan atau menguji metode pemulihan akun lainnya.
- Tidak menangani hal-hal yang digunakan dalam alur pemulihan akun seperti DNS, server email, dan penyedia telepon sebagai bagian dari perimeter keamanan penting.

Manfaat menjalankan praktik terbaik ini: Pengamanan akses ke pengguna root membangun kepercayaan bahwa tindakan di akun Anda terkontrol dan diaudit.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

AWS menawarkan alat untuk membantu mengamankan akun Anda. Namun, karena beberapa tindakan ini tidak diaktifkan secara default, Anda harus mengambil tindakan langsung untuk mengimplementasikannya. Pertimbangkan rekomendasi berikut sebagai langkah-langkah dasar untuk mengamankan Akun AWS Anda. Saat mengimplementasikan langkah-langkah ini, penting halnya untuk membangun sebuah proses penilaian dan pemantauan kontrol keamanan secara berkelanjutan.

Saat pertama kali membuat Akun AWS, Anda memulai dengan satu identitas yang memiliki akses lengkap ke semua layanan dan sumber daya AWS di akun tersebut. Identitas ini disebut pengguna root Akun AWS. Anda dapat masuk sebagai pengguna root menggunakan alamat email dan kata sandi yang digunakan untuk membuat akun. Karena tingginya tingkat akses yang diberikan untuk pengguna root AWS, Anda harus membatasi penggunaan pengguna root AWS hanya untuk tugas yang secara khusus membutuhkannya. Kredensial masuk pengguna root harus diamankan secara ketat. Selalu aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Akun AWS.

Selain alur autentikasi normal untuk masuk ke pengguna root Anda menggunakan nama pengguna, kata sandi, perangkat autentikasi multi-faktor (MFA), ada alur pemulihan akun untuk masuk ke pengguna root Akun AWS Anda dengan mengakses ke alamat email dan nomor telepon yang terkait dengan akun Anda. Oleh karena itu, pastikan Anda mengamankan akun email pengguna root yang digunakan untuk mengirimkan email pemulihan dan nomor telepon yang terkait dengan akun tersebut. Selain itu, pertimbangkan potensi rantai dependensi apabila alamat email yang terkait dengan pengguna root di-host di server email atau sumber daya layanan nama domain (DNS) dari Akun AWS yang sama.

Saat menggunakan AWS Organizations, ada beberapa Akun AWS yang masing-masing memiliki pengguna root. Satu akun ditetapkan sebagai akun manajemen dan beberapa lapisan akun anggota kemudian dapat ditambahkan di bawah akun manajemen. Prioritaskan pengamanan pengguna root di akun manajemen Anda, lalu hubungi pengguna root akun anggota Anda. Strategi pengamanan pengguna root akun manajemen Anda dapat berbeda dari pengguna root akun anggota, dan Anda dapat menerapkan kontrol keamanan preventif pada pengguna root akun anggota Anda.

Langkah implementasi

Langkah-langkah implementasi berikut direkomendasikan untuk membuat kontrol bagi pengguna root. Jika berlaku, referensi silang untuk rekomendasi dilakukan ke AWS Foundations Benchmark untuk CIS versi 1.4.0. Selain langkah-langkah ini, baca Panduan praktik terbaik AWS untuk mengamankan sumber daya dan Akun AWS Anda.

Kontrol preventif

1. Siapkan informasi kontak yang akurat untuk akun.
    a. Informasi ini digunakan untuk alur pemulihan kehilangan kata sandi, alur pemulihan kehilangan akun perangkat MFA, dan untuk komunikasi penting terkait keamanan dengan tim Anda.
    b. Gunakan alamat email yang di-host oleh domain perusahaan Anda, sebaiknya dari daftar distribusi, sebagai alamat email pengguna root Anda. Menggunakan daftar distribusi

memberikan redundansi tambahan dan keberlanjutan akses ke akun root dalam waktu lama dibanding menggunakan akun email individu.

c. Nomor telepon yang tercantum pada informasi kontak harus berupa telepon khusus dan aman untuk tujuan ini. Nomor telepon ini tidak boleh dicantumkan atau dibagikan kepada siapa pun.

2. Jangan membuat kunci akses untuk pengguna root. Jika ada kunci akses, langsung hapus (CIS 1.4).
   a. Hilangkan kredensial terprogram yang sudah lama (kunci rahasia dan akses) untuk pengguna root.
   b. Jika kunci akses pengguna root sudah ada, Anda harus mengubah proses yang menggunakan kunci tersebut untuk menggunakan kunci akses sementara dari peran AWS Identity and Access Management (IAM), kemudian hapus kunci akses pengguna root.
3. Tentukan apakah Anda perlu menyimpan kredensial untuk pengguna root.
   a. Jika Anda menggunakan AWS Organizations untuk membuat akun anggota baru, kata sandi awal untuk pengguna root pada akun anggota baru akan ditetapkan ke nilai acak yang tidak akan ditampilkan kepada Anda. Pertimbangkan untuk menggunakan alur pengaturan ulang kata sandi dari akun manajemen AWS Organization Anda untuk mendapatkan akses ke akun anggota jika diperlukan.
   b. Untuk Akun AWS atau akun AWS Organization manajemen terpisah, coba buat dan simpan kredensial dengan aman untuk pengguna root. Mengaktifkan MFA untuk pengguna root.
4. Aktifkan kontrol pencegahan untuk pengguna root akun anggota di lingkungan multiakun AWS.
   a. Pertimbangkan untuk mengaktifkan pagar pembatas preventif Jangan Izinkan Pembuatan Kunci Akses Root untuk Pengguna Root untuk akun anggota.
   b. Pertimbangkan untuk mengaktifkan pagar pembatas preventif Jangan Izinkan Tindakan sebagai Pengguna Root untuk akun anggota.
5. Jika Anda memerlukan kredensial untuk pengguna root:
   a. Gunakan kata sandi yang kompleks.
   b. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root, khususnya untuk akun manajemen (pembayar) AWS Organizations (CIS 1.5).
   c. Pertimbangkan perangkat MFA pada perangkat keras untuk ketahanan dan keamanan, karena perangkat sekali pakai dapat mengurangi kemungkinan perangkat yang berisi kode MFA Anda dapat digunakan kembali untuk tujuan lain. Pastikan baterai pada perangkat MFA perangkat keras diganti secara rutin. (CIS 1.6)

    - Untuk mengonfigurasi MFA bagi pengguna root, ikuti petunjuk untuk mengaktifkan MFA virtual atau perangkat MFA perangkat keras.

  d. Pertimbangkan untuk mendaftarkan beberapa perangkat MFA sebagai cadangan. Satu akun bisa memiliki maksimal 8 perangkat MFA.

    - Perlu diperhatikan bahwa mendaftarkan lebih dari satu perangkat MFA untuk pengguna root akan otomatis menonaktifkan alur untuk memulihkan akun Anda jika perangkat MFA hilang.

  e. Simpan kata sandi dengan aman, dan pertimbangkan dependensi melingkar jika menyimpan kata sandi secara elektronik. Jangan gunakan cara penyimpanan kata sandi yang memerlukan akses ke Akun AWS yang sama untuk mendapatkannya.

6. Opsional: Coba terapkan jadwal rotasi kata sandi untuk pengguna root secara berkala.

    - Praktik terbaik manajemen kredensial bergantung pada persyaratan peraturan dan kebijakan Anda. Pengguna root yang dilindungi oleh MFA tidak mengandalkan kata sandi sebagai satu faktor autentikasi.
    - Mengubah kata sandi pengguna root secara berkala mengurangi risiko pengungkapan kata sandi secara tidak sengaja yang dapat memicu penyalahgunaan.

Kontrol deteksi

- Buat alarm untuk mendeteksi penggunaan kredensial root (CIS 1.7). Mengaktifkan Amazon GuardDuty akan memantau dan memberikan peringatan terkait penggunaan kredensial API pengguna root melalui temuan RootCredentialUsage.
- Evaluasi dan implementasikan kontrol deteksi yang termasuk dalam Paket konformasi Pilar Keamanan AWS Well-Architected untuk AWS Config, atau jika menggunakan AWS Control Tower, kontrol yang sangat direkomendasikan dalam Control Tower.

Panduan operasional

- Tentukan siapa di organisasi Anda yang harus memiliki akses ke kredensial pengguna root.
    - Gunakan aturan dua orang sehingga tidak ada satu orang pun yang memiliki akses ke semua kredensial dan MFA yang diperlukan untuk mendapatkan akses pengguna root.
    - Pastikan bahwa organisasi, dan bukan perorangan, yang memegang kendali atas nomor telepon dan alias email yang terkait dengan akun (yang digunakan untuk alur pengaturan ulang kata sandi dan MFA).
- Gunakan pengguna root hanya untuk keperluan khusus (CIS 1.7).

  - Pengguna root AWS tidak boleh digunakan untuk tugas sehari-hari, bahkan tugas administratif. Hanya masuk sebagai pengguna root saat melakukan tugas AWS yang memerlukan pengguna root. Semua tindakan lainnya harus dilakukan oleh pengguna lain dengan peran yang sesuai.
- Periksa secara berkala apakah akses ke pengguna root berfungsi dengan baik sehingga prosedurnya telah teruji sebelum terjadi situasi darurat yang memerlukan penggunaan kredensial pengguna root.
- Periksa secara berkala apakah alamat email yang terkait dengan akun dan yang tercantum dalam Kontak Alternatif berfungsi dengan baik. Pantau kotak masuk email untuk melihat apakah ada notifikasi keamanan yang Anda terima <abuse@amazon.com>. Selain itu, pastikan semua nomor telepon yang terkait dengan akun saat ini berfungsi dengan baik.
- Siapkan prosedur respons insiden untuk merespons penyalahgunaan akun root. Lihat Panduan Respons Insiden Keamanan AWS dan praktik terbaik di bagian Respons Insiden dalam laporan resmi Pilar Keamanan untuk informasi lebih lanjut tentang membangun strategi respons insiden untuk Akun AWS Anda.

Sumber daya

Praktik Terbaik Terkait:

- SEC01-BP01 Memisahkan beban kerja menggunakan akun
- SEC02-BP01 Menggunakan mekanisme masuk yang kuat
- SEC03-BP02 Memberikan hak akses paling rendah
- SEC03-BP03 Menerapkan proses akses darurat
- SEC10-BP05 Menyediakan akses di awal

Dokumen terkait:

- AWS Control Tower
- Panduan Audit Keamanan AWS
- Praktik Terbaik IAM
- Amazon GuardDuty – peringatan penggunaan kredensial root
- Panduan langkah demi langkah tentang pemantauan untuk penggunaan kredensial root melalui CloudTrail
- Token MFA yang disetujui untuk digunakan dengan AWS

- Menerapkan akses pemicu peringatan di AWS
- 10 elemen keamanan teratas yang perlu ditingkatkan di Akun AWS Anda
- Apa yang harus saya lakukan ketika mendapati aktivitas tidak sah di akun Akun AWS saya?

Video terkait:

- Aktifkan adopsi AWS dalam skala besar dengan otomatisasi dan tata kelola
- Praktik Terbaik Keamanan dengan Cara Well-Architected
- Membatasi penggunaan kredensial root AWS dari AWS re:inforce 2022 – Praktik terbaik keamanan dengan AWS IAM

Lab dan contoh terkait:

- Lab: Akun AWS dan pengguna root

Mengoperasikan beban kerja Anda dengan aman

Mengoperasikan beban kerja dengan aman mencakup keseluruhan siklus hidup beban kerja, mulai dari merancang, membangun, menjalankan, hingga peningkatan berkelanjutan. Salah satu cara untuk meningkatkan kemampuan Anda untuk beroperasi secara aman di cloud adalah dengan mengambil pendekatan organisasional terhadap tata kelola. Tata kelola adalah bagaimana keputusan dipandu secara konsisten semata-mata atas penilaian yang baik dari orang-orang yang terlibat di dalamnya. Model dan proses tata kelola Anda adalah cara Anda menjawab pertanyaan “Bagaimana saya mengetahui bahwa sasaran kontrol untuk suatu beban kerja sudah dipenuhi dan sesuai untuk beban kerja tersebut?” Memiliki pendekatan yang konsisten dalam mengambil keputusan akan mempercepat deployment beban kerja dan membantu meningkatkan standar kemampuan keamanan dalam organisasi Anda.

Untuk mengoperasikan beban kerja dengan aman, Anda harus menerapkan praktik terbaik yang menyeluruh ke setiap area keamanan. Pilih persyaratan dan proses yang telah Anda tetapkan dalam keunggulan operasional di tingkat organisasi dan beban kerja, lalu terapkan ke semua area. Dengan terus mengikuti rekomendasi terbaru dari AWS dan industri serta kecerdasan ancaman, Anda dapat mengembangkan model ancaman dan tujuan kontrol. Mengotomatiskan proses, pengujian, dan validasi keamanan memungkinkan Anda menskalakan operasi keamanan Anda.

Dengan otomatisasi, konsistensi dan keberulangan proses dapat diwujudkan. Manusia memiliki kemampuan yang baik dalam banyak hal, tetapi melakukan hal sama secara berulang dan terus menerus tanpa kesalahan bukanlah salah satunya. Bahkan dengan buku pedoman yang jelas, tetap ada risiko ketidakkosistenan ketika orang melakukan tugas berulang. Ini dapat terjadi terutama ketika orang-orang memiliki tanggung jawab yang beragam dan mereka harus merespons peringatan yang belum familier. Namun, otomatisasi merespons dengan cara yang sama setiap kalinya. Cara terbaik untuk melakukan deployment aplikasi adalah melalui otomatisasi. Kode yang menjalankan deployment dapat diuji untuk kemudian diterapkan dalam deployment. Hal ini meningkatkan keyakinan dalam proses perubahan dan mengurangi risiko kegagalan dalam perubahan.

Untuk memverifikasi bahwa konfigurasi memenuhi sasaran kontrol Anda, uji otomatisasi dan aplikasi yang di-deploy dalam lingkungan nonproduksi terlebih dahulu. Dengan begitu, Anda dapat menguji otomatisasi untuk mengetahui apakah semua langkah telah dilakukan dengan benar. Anda juga mendapatkan umpan balik awal dalam siklus pengembangan dan deployment, meminimalkan penggarapan ulang. Untuk mengurangi peluang kesalahan deployment, lakukan perubahan konfigurasi dengan kode, bukan dengan orang. Jika Anda perlu melakukan deployment ulang sebuah aplikasi, otomatisasi akan membuat hal ini jauh lebih mudah. Begitu Anda menentukan sasaran kontrol tambahan, Anda dapat menambahkannya dengan mudah ke otomatisasi untuk semua beban kerja.

Daripada membuat setiap pemilik beban kerja menerapkan keamanan spesifik pada beban kerjanya, hemat waktu dengan menggunakan kemampuan umum dan komponen bersama. Beberapa contoh layanan yang dapat digunakan oleh banyak tim di antaranya adalah proses pembuatan akun AWS, identitas terpusat untuk orang-orang, konfigurasi pencatatan log umum, serta pembuatan gambar dasar kontainer dan AMI. Pendekatan ini dapat membantu pembangun dalam meningkatkan efisiensi waktu siklus beban kerja serta memenuhi sasaran kontrol keamanan secara konsisten. Ketika tim bekerja secara konsisten, Anda dapat memvalidasi sasaran kontrol dan membuat laporan yang lebih baik tentang postur kontrol dan posisi risiko Anda kepada pemangku kepentingan.

Praktik terbaik

- SEC01-BP03 Identifikasikan dan validasikan tujuan kontrol
- SEC01-BP04 Ikuti info terbaru tentang ancaman keamanan
- SEC01-BP05 Mengikuti informasi terbaru tentang rekomendasi keamanan
- SEC01-BP06 Mengotomatiskan pengujian dan validasi kontrol keamanan di pipeline
- SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi menggunakan model ancaman

- SEC01-BP08 Mengevaluasi dan mengimplementasikan fitur serta layanan keamanan baru secara rutin

SEC01-BP03 Identifikasikan dan validasikan tujuan kontrol

Berdasarkan persyaratan kepatuhan dan risiko yang diidentifikasi dari model ancaman Anda, dapatkan dan validasikan tujuan kontrol dan kontrol yang perlu Anda terapkan pada beban kerja Anda. Validasi berkelanjutan terhadap tujuan kontrol dan kontrol dapat membantu Anda mengukur efektivitas mitigasi risiko.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Tinggi

Panduan implementasi

- Identifikasikan persyaratan kepatuhan: Temukan persyaratan organisasi, legal, dan kepatuhan yang harus dipatuhi oleh beban kerja Anda.
- Identifikasikan sumber daya kepatuhan AWS: Identifikasikan sumber daya yang disediakan oleh AWS untuk membantu Anda dengan kepatuhan.
  - https://aws.amazon.com/compliance/
  - https://aws.amazon.com/artifact/

Sumber daya

Dokumen terkait:

- AWSPanduan Audit Keamanan
- Buletin Keamanan

Video terkait:

- AWS Security Hub: Kelola Peringatan Keamanan dan Otomatiskan Kepatuhan
- Praktik Terbaik Keamanan dengan Cara Well-Architected

SEC01-BP04 Ikuti info terbaru tentang ancaman keamanan

Kenali vektor serangan dengan terus mengikuti perkembangan ancaman keamanan terbaru untuk membantu Anda menentukan dan menerapkan kontrol yang sesuai. Gunakan AWS Managed Services untuk memudahkan Anda menerima pemberitahuan tentang perilaku yang tidak diharapkan atau tidak biasa di akun AWS Anda. Lakukan investigasi menggunakan alat Partner AWS atau feed informasi ancaman pihak ketiga sebagai bagian dari alur informasi keamanan Anda. Dengan Daftar Kerentanan dan Paparan Umum (CVE) mencantumkan kerentanan keamanan siber yang diuraikan secara publik dan dapat Anda gunakan untuk terus mengikuti perkembangan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Tinggi

Panduan implementasi

- Berlangganan ke sumber inteligensi ancaman: Tinjau informasi inteligensi ancaman secara rutin dari berbagai sumber yang relevan dengan teknologi yang digunakan di beban kerja Anda.
  - Daftar Kerentanan dan Paparan Umum
- Pertimbangkan layanan AWS Shield Advanced : Layanan ini menyediakan visibilitas waktu nyata ke sumber inteligensi, jika beban kerja Anda dapat diakses internet.

Sumber daya

Dokumen terkait:

- Panduan Audit Keamanan AWS
- AWS Shield
- Buletin Keamanan

Video terkait:

- Security Best Practices the Well-Architected Way

SEC01-BP05 Mengikuti informasi terbaru tentang rekomendasi keamanan

Ikuti terus rekomendasi keamanan AWS dan industri untuk mengembangkan postur keamanan beban kerja Anda. Buletin Keamanan AWS berisi informasi penting tentang notifikasi keamanan dan privasi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

- Ikuti informasi terbaru AWS: Lakukan langganan atau kunjungi secara rutin untuk mendapatkan saran, tips, dan trik baru.
  - Lab AWS Well-Architected
  - Blog keamanan AWS
  - Dokumentasi layanan AWS
- Lakukan langganan ke berita-berita industri: Rutin tinjau umpan berita dari berbagai sumber terkait teknologi yang digunakan di beban kerja Anda.
  - Contoh: Daftar Kerentanan dan Paparan Umum

Sumber daya

Dokumen terkait:

- Buletin Keamanan

Video terkait:

- Praktik Terbaik Keamanan dengan Cara Well-Architected

SEC01-BP06 Mengotomatiskan pengujian dan validasi kontrol keamanan di pipeline

Tetapkan acuan dasar (baseline) dan templat yang aman untuk mekanisme keamanan yang telah diuji dan divalidasi sebagai bagian dari build, pipeline, dan proses Anda. Gunakan alat dan otomatisasi untuk menguji dan memvalidasi semua kontrol keamanan secara terus-menerus. Misalnya, pindai item seperti image mesin dan templat infrastruktur sebagai kode (IaC) untuk menemukan kerentanan keamanan, kejanggalan, dan penyimpangan dari acuan dasar yang telah ditetapkan pada setiap tahap. AWS CloudFormation Guard dapat membantu Anda memverifikasi bahwa templat CloudFormation aman, menghemat waktu, dan mengurangi risiko kesalahan konfigurasi.

Mengurangi jumlah kesalahan konfigurasi keamanan yang dimasukkan ke dalam lingkungan produksi adalah hal penting—makin banyak kontrol kualitas dan pengurangan kecacatan yang dapat Anda lakukan dalam proses build, makin baik hasilnya. Rancang pipeline integrasi berkelanjutan dan deployment berkelanjutan (CI/CD) untuk menguji masalah keamanan setiap kali memungkinkan. Pipeline CI/CD menawarkan kesempatan untuk menyempurnakan keamanan pada tiap-tiap tahap build dan pengiriman. Peralatan keamanan CI/CD juga harus terus diperbarui untuk memitigasi ancaman yang berkembang.

Lacak perubahan pada konfigurasi beban kerja Anda untuk membantu audit kepatuhan, manajemen perubahan, dan penyelidikan yang mungkin berlaku untuk Anda. Anda dapat menggunakan AWS Config untuk merekam dan mengevaluasi sumber daya AWS dan pihak ketiga Anda. Ini memungkinkan Anda untuk mengaudit dan menilai secara berkelanjutan keseluruhan kepatuhan terhadap aturan dan paket kesesuaian, yakni kumpulan aturan dengan tindakan perbaikan.

Pelacakan perubahan harus menyertakan perubahan terencana, yang merupakan bagian dari proses kontrol perubahan organisasi Anda (terkadang disebut MACD—Memindah, Menambah, Mengubah, Menghapus), perubahan tidak terencana, dan perubahan yang tidak diinginkan, seperti insiden. Perubahan dapat terjadi pada infrastruktur, tetapi mungkin juga berkaitan dengan kategori lain, seperti perubahan pada repositori kode, perubahan image mesin dan inventaris aplikasi, perubahan proses dan kebijakan, atau perubahan dokumentasi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

- Otomatiskan manajemen konfigurasi: Tegakkan dan validasi konfigurasi keamanan secara otomatis menggunakan layanan atau alat manajemen konfigurasi.
  - AWS Systems Manager
  - AWS CloudFormation
  - Menyiapkan Pipeline CI/CD di AWS

Sumber daya

Dokumen terkait:

- Cara menggunakan kebijakan kontrol layanan untuk menetapkan pagar pembatas izin di seluruh akun dalam Organisasi AWS Anda

Video terkait:

- Mengelola Lingkungan AWS Multiakun Menggunakan AWS Organizations
- Praktik Terbaik Keamanan dengan Cara Well-Architected

SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi menggunakan model ancaman

Lakukan pemodelan ancaman untuk mengidentifikasi dan menyediakan daftar potensi ancaman terbaru serta mitigasi terkait untuk beban kerja Anda. Tentukan prioritas ancaman dan sesuaikan mitigasi kontrol keamanan Anda untuk mencegah, mendeteksi, dan merespons ancaman. Periksa kembali dan kelola sesuai dengan konteks beban kerja Anda, serta lanskap keamanan yang terus berubah.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Apa itu pemodelan ancaman?

“Pemodelan ancaman digunakan untuk mengidentifikasi, mengomunikasikan, serta memahami ancaman dan mitigasi untuk melindungi suatu nilai.” – The Open Web Application Security Project (OWASP) Application Threat Modeling

Mengapa pemodelan ancaman sebaiknya dilakukan?

Sistem begitu kompleks. Kompleksitas dan kemampuannya akan makin meningkat seiring waktu, sehingga memberikan nilai bisnis yang lebih banyak dan meningkatkan keterlibatan serta kepuasan pelanggan. Artinya, keputusan desain IT perlu mempertimbangkan peningkatan jumlah kasus penggunaan. Perubahan kompleksitas dan jumlah kasus penggunaan ini biasanya menjadikan pendekatan tidak terstruktur tidak efektif untuk menemukan dan memitigasi ancaman. Maka, Anda memerlukan pendekatan sistematis untuk menghitung potensi ancaman terhadap sistem, serta untuk melakukan dan memprioritaskan mitigasi untuk memastikan organisasi Anda dapat meningkatkan postur keamanan sistem secara keseluruhan dengan maksimal meski dengan sumber daya terbatas.

Pemodelan ancaman dirancang untuk memberikan pendekatan sistematis ini, bertujuan untuk menemukan dan mengatasi masalah dalam proses desain lebih awal, saat biaya dan upaya mitigasi relatif rendah dibandingkan setelahnya dalam siklus hidup. Pendekatan ini selaras dengan prinsip industri keamanan shift-left. Pada intinya, pemodelan ancaman akan terintegrasi dengan

proses manajemen risiko organisasi serta membantu menentukan kontrol mana yang akan diimplementasikan menggunakan pendekatan menurut ancaman.

Kapan seharusnya pemodelan ancaman dilakukan?

Mulai pemodelan ancaman dalam siklus hidup beban kerja Anda sedini mungkin. Hal ini membuat Anda lebih fleksibel dalam menentukan tindakan untuk mengatasi ancaman yang teridentifikasi. Seperti halnya bug perangkat lunak, makin dini Anda mengidentifikasi ancaman, makin hemat biaya penanganannya. Model ancaman adalah dokumen hidup (living document) dan akan terus berkembang seiring perubahan beban kerja Anda. Periksa kembali model ancaman Anda dari waktu ke waktu, termasuk saat terjadi perubahan besar, perubahan dalam lanskap ancaman, atau saat mengadopsi fitur atau layanan baru.

Langkah implementasi

Bagaimana cara menjalankan pemodelan ancaman?

Pemodelan ancaman bisa dijalankan dengan banyak cara. Seperti halnya bahasa pemrograman, setiap model memiliki kelebihan dan kekurangannya masing-masing. Pilih cara yang paling tepat untuk Anda. Salah satu pendekatannya adalah memulai dengan Shostack's 4 Question Frame for Threat Modeling yang berisi pertanyaan terbuka agar pengujian pemodelan ancaman Anda terstruktur:

1. Apa yang sedang Anda kerjakan?

   Pertanyaan ini bertujuan untuk membantu memahami dan menentukan sistem yang sedang Anda bangun serta detail sistem tersebut yang relevan dengan keamanan. Membuat model atau diagram adalah salah satu cara paling populer untuk menjawab pertanyaan ini karena dapat membantu Anda memvisualisasikan apa yang sedang Anda bangun, misalnya, menggunakan diagram alur data. Menuliskan asumsi dan detail penting tentang sistem Anda juga dapat membantu Anda menentukan cakupan. Hal ini membantu menyatukan fokus semua orang yang berkontribusi dalam model ancaman, serta menghindari topik di luar cakupan (termasuk versi lama sistem Anda) yang memakan banyak waktu. Misalnya, jika Anda sedang membangun aplikasi web, sepertinya tidak perlu membuang waktu melakukan pemodelan ancaman untuk urutan boot tepercaya sistem operasi untuk klien browser karena desain Anda tidak akan memengaruhi hal ini.
2. Apa saja potensi permasalahannya?

   Di sini Anda dapat mengidentifikasi ancaman terhadap sistem Anda. Ancaman adalah tindakan atau peristiwa yang disengaja atau tidak disengaja, yang dampaknya tidak diharapkan dan

dapat memengaruhi keamanan sistem Anda. Tanpa mengetahui dengan jelas apa saja potensi permasalahannya, Anda tidak akan tahu cara penanganannya.

Tidak ada daftar khusus tentang apa saja masalah yang dapat terjadi. Pembuatan daftar ini memerlukan diskusi dan kolaborasi antara setiap individu di dalam tim Anda serta pihak terkait yang terlibat dalam pengujian pemodelan ancaman. Anda dapat melengkapi diskusi dengan model untuk mengidentifikasi ancaman, seperti STRIDE, yang menunjukkan berbagai kategori evaluasi: Penipuan (Spoofing), Gangguan (Tampering), Penolakan (Repudiation), Kebocoran Informasi (Information Disclosure), Penolakan Layanan (Denial of Service), dan Peningkatan Hak Istimewa (Elevation of Privilege). Selain itu, Anda mungkin ingin melengkapi diskusi dengan meninjau daftar yang sudah ada dan penelitian untuk inspirasi, termasuk OWASP Top 10, HiTrust Threat Catalog, dan katalog ancaman milik organisasi Anda.

3. Apa tindakan yang akan Anda lakukan?

   Sama seperti pertanyaan sebelumnya, tidak ada daftar khusus untuk semua kemungkinan mitigasi. Input dalam langkah ini adalah ancaman yang teridentifikasi, pelaku, dan area peningkatan dari langkah sebelumnya.

   Keamanan dan kepatuhan adalah tanggung jawab bersama antara Anda dan AWS. Perlu dipahami bahwa pertanyaan “Tindakan apa yang akan kita lakukan?” harus disertai dengan pertanyaan “Siapa yang bertanggung untuk melakukan hal ini?”. Memahami keseimbangan tanggung jawab antara Anda dan AWS membantu Anda menentukan cakupan pengujian pemodelan ancaman ke mitigasi dalam kontrol Anda, yang biasanya merupakan gabungan dari opsi konfigurasi layanan AWS dan mitigasi dari sistem Anda sendiri.

   Untuk porsi tanggung jawab bersama AWS, Anda akan melihat bahwa layanan AWS masuk dalam cakupan banyak program kepatuhan. Program-program tersebut akan membantu Anda memahami penerapan kontrol yang andal di AWS untuk menjaga keamanan dan kepatuhan cloud. Laporan audit dari program-program ini dapat diunduh oleh pelanggan AWS melalui AWS Artifact.

   Apa pun layanan AWS yang Anda gunakan, selalu ada elemen yang menjadi tanggung jawab pelanggan, dan mitigasi yang diselaraskan dengan tanggung jawab ini harus disertakan dalam model ancaman Anda. Untuk mitigasi kontrol keamanan bagi layanan AWS sendiri, Anda perlu mempertimbangkan implementasi kontrol keamanan di seluruh domain, termasuk domain seperti manajemen akses dan identitas (autentikasi dan otorisasi), perlindungan data (diam dan bergerak), keamanan infrastruktur, pencatatan log, dan pemantauan. Dokumentasi untuk setiap layanan AWS memiliki bab keamanan khusus yang menyediakan panduan tentang kontrol keamanan yang perlu dipertimbangkan sebagai mitigasi. Pertimbangkan kode yang Anda tulis

dan dependensi kodenya karena hal ini sangat penting, serta tentukan kontrol yang dapat Anda terapkan untuk mengatasi ancaman. Kontrol ini dapat berupa validasi input, penanganan sesi, dan penanganan ikatan. Fokus pada kode kustom karena sebagian besar kerentanan seringnya terjadi di area ini.

4. Apakah kita melakukannya dengan baik?

   Tujuannya adalah agar tim dan organisasi Anda dapat meningkatkan kualitas model ancaman dan kecepatan dalam melakukan pemodelan ancaman dari waktu ke waktu. Peningkatan ini adalah hasil dari gabungan praktik, pembelajaran, pengajaran, dan peninjauan. Agar Anda dan tim Anda dapat mempelajari lebih lanjut dan melakukan praktik langsung, sebaiknya ikuti lokakarya atau kursus pelatihan Pemodelan ancaman yang benar bagi pembangun. Selain itu, jika Anda mencari panduan tentang cara mengintegrasikan pemodelan ancaman ke dalam siklus hidup pengembangan organisasi Anda, lihat posting Cara memanfaatkan pemodelan ancaman di halaman AWS Security Blog.

Komposer Ancaman

Untuk membantu dan memandu Anda dalam membuat model ancaman, pertimbangkan untuk menggunakan alat Komposer Ancaman, yang bertujuan untuk mempercepat waktu untuk mencapai nilai saat membuat model ancaman. Alat ini membantu Anda melakukan hal berikut:

- Menulis pernyataan ancaman berguna yang selaras dengan tata bahasa ancaman yang bekerja dalam alur kerja non-linear alami
- Menghasilkan model ancaman yang dapat dibaca manusia
- Membuat model ancaman yang dapat dibaca mesin untuk memungkinkan Anda memperlakukan model ancaman sebagai kode
- Membantu Anda mengidentifikasi area peningkatan kualitas dan cakupan dengan cepat menggunakan Dasbor Wawasan

Untuk referensi lebih lanjut, kunjungi Komposer Ancaman dan beralih ke Ruang Kerja Contoh yang ditentukan sistem.

Sumber daya

Praktik Terbaik Terkait:

- SEC01-BP03 Identifikasikan dan validasikan tujuan kontrol

- SEC01-BP04 Ikuti info terbaru tentang ancaman keamanan
- SEC01-BP05 Mengikuti informasi terbaru tentang rekomendasi keamanan
- SEC01-BP08 Mengevaluasi dan mengimplementasikan fitur serta layanan keamanan baru secara rutin

Dokumen terkait:

- Cara memanfaatkan pemodelan ancaman (AWS Security Blog)
- NIST: Panduan untuk Pemodelan Ancaman Sistem yang Terpusat pada Data

Video terkait:

- AWS Summit ANZ 2021 - Cara memanfaatkan pemodelan ancaman
- AWS Summit ANZ 2022 - Menskalakan keamanan – Pengoptimalan untuk pengiriman yang cepat dan aman

Pelatihan terkait:

- Pemodelan ancaman yang benar bagi pembangun – Pelatihan mandiri virtual AWS Skill Builder
- Pemodelan ancaman yang benar bagi pembangun – AWS Workshop

Alat terkait:

- Komposer Ancaman

SEC01-BP08 Mengevaluasi dan mengimplementasikan fitur serta layanan keamanan baru secara rutin

Evaluasikan dan implementasikan fitur serta layanan keamanan dari Partner AWS dan AWS yang memungkinkan peningkatan postur keamanan beban kerja. Blog Keamanan AWS menyoroti fitur dan layanan baru AWS, panduan implementasi, dan panduan keamanan umum. Yang Baru dengan AWS? adalah cara terbaik untuk tetap mengetahui fitur, layanan, dan pengumuman baru dari AWS.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Rencanakan peninjauan rutin: Buat kalender aktivitas peninjauan yang mencakup persyaratan kepatuhan, evaluasi fitur dan layanan keamanan baru AWS, serta tetap mengikuti berita terbaru industri.
- Temukan fitur dan layanan AWS: Temukan fitur keamanan yang tersedia untuk layanan yang Anda gunakan, dan tinjau fitur baru setelah dirilis.
  - Blog keamanan AWS
  - Buletin keamanan AWS
  - Dokumentasi layanan AWS
- Tentukan proses onboarding layanan AWS: Tentukan proses untuk onboarding layanan baru AWS. Sertakan cara Anda mengevaluasi layanan baru AWS untuk fungsionalitas, dan persyaratan kepatuhan untuk beban kerja Anda.
- Uji coba fitur dan layanan baru: Uji coba fitur dan layanan baru setelah dirilis di lingkungan nonproduksi yang direplikasi menyerupai lingkungan produksi Anda.
- Implementasikan mekanisme pertahanan lainnya: Implementasikan mekanisme otomatis untuk melindungi beban kerja, dan menelusuri opsi yang tersedia.
  - Mengatasi sumber daya AWS yang tidak patuh dengan Aturan AWS Config

Sumber daya

Video terkait:

- Praktik Terbaik Keamanan dengan Cara Well-Architected

Manajemen identitas dan akses

Untuk menggunakan layanan AWS, Anda harus memberikan akses ke sumber daya di akun AWS Anda kepada pengguna dan aplikasi. Seiring dengan bertambahnya beban kerja yang Anda jalankan di AWS, Anda perlu menerapkan izin dan manajemen identitas yang andal guna memastikan orang-orang yang tepatlah yang mendapatkan akses ke sumber daya yang tepat dengan persyaratan yang tepat. AWS menawarkan beragam pilihan kemampuan untuk membantu Anda mengelola identitas mesin dan manusia serta izin mereka. Praktik terbaik untuk kemampuan ini termasuk dalam dua area utama.

Topik

- Manajemen identitas
- Manajemen izin

Manajemen identitas

Ada dua jenis identitas yang harus Anda kelola ketika menentukan pendekatan terhadap pengoperasian beban kerja AWS yang aman.

- Identitas manusia: Administrator, developer, operator, dan konsumen aplikasi Anda memerlukan identitas untuk mengakses lingkungan dan aplikasi AWS. Identitas ini mencakup anggota organisasi Anda, atau pengguna eksternal yang berkolaborasi dengan Anda, dan yang berinteraksi dengan sumber daya AWS Anda melalui browser web, aplikasi klien, atau alat baris perintah interaktif.
- Identitas mesin: Aplikasi beban kerja, alat operasional, dan komponen Anda memerlukan identitas untuk membuat permintaan ke layanan AWS, misalnya, untuk membaca data. Identitas ini mencakup mesin yang dijalankan di lingkungan AWS Anda, seperti instans Amazon EC2 atau fungsi AWS Lambda. Anda juga dapat mengelola identitas mesin untuk pihak eksternal yang membutuhkan akses. Selain itu, Anda mungkin juga memiliki mesin di luar AWS yang memerlukan akses ke lingkungan AWS Anda.

Praktik terbaik

- SEC02-BP01 Menggunakan mekanisme masuk yang kuat
- SEC02-BP02 Menggunakan kredensial sementara
- SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman

- SEC02-BP04 Mengandalkan penyedia identitas terpusat
- SEC02-BP05 Mengaudit dan merotasi kredensial secara berkala
- SEC02-BP06 Manfaatkan grup dan atribut pengguna

SEC02-BP01 Menggunakan mekanisme masuk yang kuat

Proses masuk (autentikasi menggunakan kredensial masuk) dapat menimbulkan risiko jika tidak menggunakan mekanisme seperti autentikasi multi-faktor (MFA), khususnya ketika kredensial tanpa sengaja bocor atau mudah ditebak. Untuk mengurangi risiko ini, gunakan mekanisme masuk yang kuat dengan menerapkan MFA dan kebijakan kata sandi yang kuat.

Hasil yang diinginkan: Mengurangi risiko adanya akses yang tidak diinginkan ke kredensial di AWS menggunakan mekanisme masuk yang kuat bagi pengguna AWS Identity and Access Management (IAM), pengguna root Akun AWS, AWS IAM Identity Center (pengganti AWS Single Sign-On), dan penyedia identitas pihak ketiga. Tujuan ini dapat dicapai dengan MFA, menerapkan kebijakan kata sandi yang kuat, dan mendeteksi perilaku masuk tidak wajar.

Antipola umum:

- Tidak menerapkan kebijakan kata sandi yang kuat untuk identitas Anda, termasuk kata sandi yang kompleks dan MFA.
- Kredensial yang sama digunakan oleh pengguna yang berbeda.
- Tidak menggunakan kontrol deteksi untuk aktivitas masuk yang mencurigakan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Ada banyak cara bagi identitas manusia masuk untuk ke AWS. Ini adalah praktik terbaik AWS untuk mengandalkan penyedia identitas terpusat menggunakan federasi (federasi langsung atau dengan AWS IAM Identity Center) saat melakukan autentikasi ke AWS. Dalam kasus tersebut, Anda harus membuat proses masuk yang aman dengan penyedia identitas atau Microsoft Active Directory.

Saat pertama kali membuka Akun AWS, Anda akan memulainya dengan pengguna root Akun AWS. Hanya gunakan akun pengguna root untuk mengonfigurasikan akses bagi pengguna Anda (dan untuk tugas yang memerlukan pengguna root). Penting halnya untuk segera mengaktifkan MFA bagi akun pengguna root setelah membuka Akun AWS Anda dan mengamankan pengguna root menggunakan panduan praktik terbaik AWS.

Jika Anda membuat pengguna di AWS IAM Identity Center, amankan proses masuk dalam layanan tersebut. Untuk identitas konsumen, Anda dapat menggunakan Amazon Cognito user pools dan mengamankan proses masuk dalam layanan tersebut, atau dengan salah satu penyedia identitas yang didukung Amazon Cognito user pools.

Jika Anda menggunakan pengguna AWS Identity and Access Management (IAM), amankan proses masuk menggunakan IAM.

Apa pun metode masuknya, menerapkan kebijakan masuk yang kuat adalah hal yang sangat penting.

Langkah implementasi

Berikut ini adalah rekomendasi mekanisme masuk yang kuat secara umum. Pengaturan aktual yang Anda konfigurasikan harus diatur sesuai kebijakan perusahaan Anda atau gunakan standar seperti NIST 800-63.

- Terapkan MFA. Ini adalah praktik terbaik IAM untuk menerapkan MFA untuk beban kerja dan identitas manusia. Mengaktifkan MFA akan memberikan lapisan keamanan tambahan yang mengharuskan pengguna untuk memberikan kredensial masuk dan kata sandi sekali pakai (OTP) atau string yang dibuat dan diverifikasi secara kriptografik dari perangkat untuk perangkat keras.
- Terapkan batas minimum karakter kata sandi, yang merupakan faktor utama dari kekuatan kata sandi.
- Terapkan kompleksitas kata sandi agar tidak mudah ditebak.
- Izinkan pengguna mengubah kata sandi mereka sendiri.
- Buat identitas individu, bukan kredensial bersama. Dengan membuat identitas individu, Anda dapat memberikan kredensial keamanan yang unik kepada setiap pengguna. Pengguna individu juga memungkinkan pengauditan aktivitas setiap pengguna.

Rekomendasi IAM Identity Center:

- IAM Identity Center memberikan kebijakan kata sandi yang telah ditentukan sebelumnya apabila menggunakan direktori default yang menerapkan persyaratan jumlah karakter, kompleksitas, dan penggunaan ulang kata sandi.
- Aktifkan MFA dan konfigurasikan pengaturan sesuai konteks (context-aware) atau selalu aktif (always-on) untuk MFA saat sumber identitas merupakan AWS Managed Microsoft AD, AD Connector, atau direktori default.

- Izinkan pengguna untuk mendaftarkan perangkat MFA miliknya.

Rekomendasi direktori Amazon Cognito user pools:

- Konfigurasikan pengaturan Kekuatan kata sandi.
- Terapkan MFA bagi pengguna.
- Gunakan pengaturan keamanan lanjutan Amazon Cognito user pools untuk fitur seperti autentikasi adaptif yang dapat memblokir aktivitas masuk yang mencurigakan.

Rekomendasi pengguna IAM:

- Idealnya, Anda menggunakan IAM Identity Center atau federasi langsung. Namun, Anda mungkin membutuhkan pengguna IAM. Dalam kasus seperti itu, atur kebijakan kata sandi untuk pengguna IAM. Anda dapat menggunakan kebijakan kata sandi untuk menentukan persyaratan, seperti minimum karakter atau apakah kata sandi harus terdiri dari karakter nonalfabet.
- Buat kebijakan IAM untuk menerapkan mekanisme masuk MFA sehingga pengguna dapat mengelola perangkat MFA dan kata sandi miliknya.

Sumber daya

Praktik Terbaik Terkait:

- SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman
- SEC02-BP04 Mengandalkan penyedia identitas terpusat
- SEC03-BP08 Membagikan sumber daya secara aman dalam organisasi Anda

Dokumen terkait:

- Kebijakan Kata Sandi AWS IAM Identity Center (pengganti AWS Single Sign-On)
- Kebijakan kata sandi pengguna IAM
- Mengatur kata sandi pengguna root Akun AWS
- Kebijakan kata sandi Amazon Cognito
- Kredensial AWS
- Praktik terbaik keamanan IAM

Video terkait:

- Mengelola izin pengguna dalam skala besar dengan AWS IAM Identity Center
- Menguasai identitas di setiap lapisan susunan

SEC02-BP02 Menggunakan kredensial sementara

Saat melakukan autentikasi jenis apa pun, sebaiknya gunakan kredensial sementara daripada kredensial jangka panjang untuk mengurangi atau menghindari risiko seperti pengungkapan, pembagian, dan pencurian kredensial.

Hasil yang diinginkan: Untuk mengurangi risiko kredensial jangka panjang, sebisa mungkin gunakan kredensial sementara untuk identitas mesin dan manusia. Kredensial jangka panjang menimbulkan banyak risiko, misalnya, dapat diunggah ke repositori GitHub publik dalam bentuk kode. Dengan kredensial sementara, Anda dapat secara signifikan mengurangi risiko penyusupan kredensial.

Antipola umum:

- Developer memilih menggunakan kunci akses jangka panjang dari IAM users dibanding memperoleh kredensial sementara dari CLI menggunakan federasi.
- Developer menyematkan kunci akses jangka panjang dalam kodenya dan mengunggah kode tersebut ke repositori Git publik.
- Developer menyematkan kunci akses jangka panjang di aplikasi seluler yang kemudian dibuat tersedia di toko aplikasi.
- Pengguna membagikan kunci akses jangka panjang kepada pengguna lainnya, atau karyawan yang sudah keluar dari perusahaan tetapi masih memiliki kunci akses jangka panjang.
- Menggunakan kunci akses jangka panjang untuk identitas mesin meski kredensial sementara dapat digunakan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Gunakan kredensial keamanan sementara, bukan kredensial jangka panjang untuk semua permintaan CLI dan API AWS. Permintaan CLI dan API ke layanan AWS harus, hampir di setiap kasus, ditandatangani menggunakan kunci akses AWS. Permintaan ini dapat ditandatangani dengan kredensial jangka panjang maupun sementara. Satu-satunya situasi yang perlu menggunakan

kredensial jangka panjang, disebut juga kunci akses jangka panjang, adalah ketika Anda menggunakan pengguna IAM atau pengguna root Akun AWS. Saat Anda bergabung ke AWS atau mengambil peran IAM melalui metode lainnya, kredensial sementara akan dibuat. Bahkan ketika Anda mengakses AWS Management Console menggunakan kredensial masuk, kredensial sementara akan dibuat untuk Anda untuk melakukan panggilan ke layanan AWS. Anda hanya memerlukan kredensial jangka panjang untuk beberapa situasi saja; hampir semua tugas dapat dilakukan menggunakan kredensial sementara.

Menghindari penggunaan kredensial jangka panjang dan mengutamakan kredensial sementara harus diikuti dengan penerapan strategi pengurangan penggunaan pengguna IAM untuk mengutamakan federasi dan peran IAM. Meski sebelumnya pengguna IAM sudah digunakan untuk identitas mesin dan manusia, kini sebaiknya jangan gunakan pengguna tersebut untuk menghindari risiko dalam penggunaan kunci akses jangka panjang.

Langkah implementasi

Untuk identitas manusia seperti karyawan, administrator, developer, operator, dan pelanggan:

- Anda harus mengandalkan penyedia identitas terpusat dan dan mengharuskan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensial sementara. Federasi untuk pengguna Anda dapat dilakukan dengan federasi langsung ke setiap Akun AWS atau menggunakan AWS IAM Identity Center (pengganti AWS IAM Identity Center) dan penyedia identitas yang Anda pilih. Selain mengurangi penggunaan kredensial jangka panjang, federasi memberikan berbagai manfaat atas penggunaan pengguna IAM. Pengguna Anda juga dapat meminta kredensial sementara dari baris perintah untuk federasi langsung atau menggunakan IAM Identity Center. Artinya, ada beberapa kasus penggunaan yang memerlukan kredensial jangka panjang atau pengguna IAM untuk pengguna Anda.
- Saat memberi pihak ketiga, seperti penyedia perangkat lunak sebagai layanan (SaaS), akses ke sumber daya di Akun AWS Anda, Anda dapat menggunakan peran lintas akun dan kebijakan berbasis sumber daya.
- Jika Anda perlu memberi aplikasi untuk konsumen atau pelanggan akses ke sumber daya AWS Anda, Anda dapat menggunakan kolam identitas Amazon Cognito atau Amazon Cognito user pools untuk menyediakan kredensial sementara. Izin untuk kredensial ini dikonfigurasikan lewat peran IAM. Anda juga dapat menentukan peran IAM terpisah dengan izin terbatas untuk pengguna tamu yang tidak diautentikasi.

Untuk identitas mesin, Anda mungkin perlu menggunakan kredensial jangka panjang. Dalam kasus tersebut, Anda harus mewajibkan beban kerja untuk menggunakan kredensial sementara dengan peran IAM untuk mengakses AWS.

- Untuk Amazon Elastic Compute Cloud (Amazon EC2), Anda dapat menggunakan peran untuk Amazon EC2.

- AWS Lambda memungkinkan Anda untuk mengonfigurasikan peran eksekusi Lambda guna memberikan izin layanan untuk melakukan tindakan AWS menggunakan kredensial sementara. Ada banyak model serupa untuk layanan AWS yang digunakan untuk memberikan kredensial sementara menggunakan peran IAM.
- Untuk perangkat IoT, Anda dapat menggunakan penyedia kredensial AWS IoT Core untuk meminta kredensial sementara.
- Untuk sistem on-premise atau sistem yang berjalan di luar AWS yang memerlukan akses ke sumber daya AWS, Anda dapat menggunakan IAM Roles Anywhere.

Dalam beberapa skenario, kredensial sementara tidak dapat digunakan dan Anda mungkin perlu menggunakan kredensial jangka panjang. Dalam situasi tersebut, audit dan rotasikan kredensial secara berkala serta rotasikan kunci akses secara rutin untuk kasus penggunaan yang memerlukan kredensial jangka panjang. Beberapa contoh yang dapat mengharuskan kredensial jangka panjang termasuk plugin WordPress dan klien pihak ketiga AWS. Dalam situasi yang mengharuskan Anda menggunakan kredensial jangka panjang, atau untuk kredensial selain kunci akses AWS, seperti masuk ke basis data, Anda dapat menggunakan layanan yang dirancang untuk menangani manajemen rahasia, seperti AWS Secrets Manager. Secrets Manager memudahkan manajemen, rotasi, dan penyimpanan rahasia terenkripsi dengan aman menggunakan layanan yang didukung. Untuk informasi lebih lanjut tentang merotasi kredensial jangka panjang, lihat merotasi kunci akses.

Sumber daya

Praktik Terbaik Terkait:

- SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman
- SEC02-BP04 Mengandalkan penyedia identitas terpusat
- SEC03-BP08 Membagikan sumber daya secara aman dalam organisasi Anda

Dokumen terkait:

- Kredensial Keamanan Sementara
- Kredensial AWS
- Praktik Terbaik Keamanan IAM
- Peran IAM
- IAM Identity Center
- Penyedia Identitas dan Federasi
- Merotasi Kunci Akses
- Solusi Partner Keamanan: Akses dan Kontrol Akses
- Pengguna Root Akun AWS

Video terkait:

- Mengelola izin pengguna dalam skala besar dengan AWS IAM Identity Center (pengganti AWS IAM Identity Center)
- Menguasai identitas di setiap lapisan susunan

SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman

Beban kerja memerlukan kemampuan otomatis untuk membuktikan identitasnya ke basis data, sumber daya, dan layanan pihak ketiga. Hal ini dapat dilakukan menggunakan kredensial akses rahasia, seperti kunci akses API, kata sandi, dan token OAuth. Menggunakan layanan yang dibuat khusus untuk menyimpan, mengelola, dan merotasi kredensial ini membantu mengurangi kemungkinan penyusupan kredensial.

Hasil yang diinginkan: Mengimplementasikan mekanisme untuk mengelola kredensial aplikasi secara aman, yang mencapai tujuan berikut:

- Mengidentifikasi rahasia apa yang diperlukan untuk beban kerja.
- Mengurangi kebutuhan kredensial jangka panjang yang diperlukan dan menggunakan kredensial jangka pendek jika memungkinkan.
- Membangun penyimpanan yang aman dan rotasi otomatis untuk kredensial jangka panjang yang tersisa.
- Mengaudit akses ke rahasia yang ada di beban kerja.

- Pemantauan berkelanjutan untuk memverifikasi bahwa tidak ada rahasia yang disematkan di kode sumber selama proses pengembangan.
- Mengurangi kemungkinan pengungkapan kredensial secara tidak sengaja.

Antipola umum:

- Tidak merotasi kredensial.
- Menyimpan kredensial jangka panjang dalam kode sumber atau file konfigurasi.
- Menyimpan kredensial diam tanpa dienkripsi.

Manfaat menjalankan praktik terbaik ini:

- Rahasia yang disimpan diamankan dengan enkripsi saat diam maupun bergerak.
- Akses ke kredensial harus melewati API (bayangkan ini seperti mesin penjual otomatis kredensial).
- Akses ke kredensial (baca dan tulis) diaudit dan dicatat.
- Pemisahan masalah (Separation of concerns): rotasi kredensial dilakukan oleh komponen terpisah, yang dapat dipisahkan dari bagian arsitektur lainnya.
- Rahasia didistribusikan secara otomatis ke komponen perangkat lunak sesuai permintaan dan rotasi dilakukan di lokasi pusat.
- Akses ke kredensial dapat dikontrol dengan sangat ketat.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Dahulu, kredensial digunakan untuk mengautentikasi basis data, API pihak ketiga, token, dan rahasia lainnya yang mungkin disematkan dalam kode sumber atau dalam file lingkungan. AWS menyediakan beberapa mekanisme untuk menyimpan kredensial ini secara aman, merotasinya secara otomatis, dan mengaudit penggunaannya.

Cara terbaik untuk mengelola rahasia adalah dengan mengikuti panduan penghapusan, penggantian, dan rotasi. Kredensial yang paling aman adalah kredensial yang tidak perlu Anda simpan, kelola, atau tangani. Jika ada kredensial yang sudah tidak digunakan untuk menjalankan beban kerja, Anda dapat menghapusnya.

Apabila kredensial masih diperlukan untuk menjalankan beban kerja dengan benar, kredensial jangka panjangnya mungkin bisa diganti dengan kredensial sementara atau jangka pendek. Misalnya, daripada melakukan hard-coding kunci akses rahasia AWS, coba ganti kredensial jangka panjang tersebut dengan kredensial sementara menggunakan peran IAM.

Beberapa rahasia yang sudah lama ada mungkin tidak dapat dihapus atau diganti. Rahasia tersebut dapat disimpan dalam layanan seperti AWS Secrets Manager, tempat rahasia bisa disimpan, dikelola, dan dirotasi secara rutin dan terpusat.

Pengauditan file konfigurasi dan kode sumber beban kerja dapat menunjukkan berbagai jenis kredensial. Tabel berikut merangkum strategi yang digunakan untuk menangani jenis kredensial umum:

Credential type	Description	Suggested strategy
IAM access keys	AWS IAM access and secret keys used to assume IAM roles inside of a workload	Replace: Use Peran IAM assigned to the compute instances (such as Amazon EC2 or AWS Lambda) instead. For interoperability with third parties that require access to resources in your Akun AWS, ask if they support Akses lintas akun AWS. For mobile apps, consider using temporary credentials through Kolam identitas (identitas gabungan) Amazon Cognito. For workloads running outside of AWS, consider IAM Roles Anywhere or AWS Systems Manager Hybrid Activations.
SSH keys	Secure Shell private keys used to log into Linux EC2 instances, manually or as part of an automated process	Replace: Use AWS Systems Manager or EC2 Instance Connect to provide programmatic and human

Credential type	Description	Suggested strategy
		access to EC2 instances using IAM roles.
Application and database credentials	Passwords – plain text string	Rotate: Store credentials in AWS Secrets Manager and establish automated rotation if possible.
Amazon RDS and Aurora Admin Database credentials	Passwords – plain text string	Replace: Use the Integrasi Secrets Manager dengan Amazon RDS or Amazon Aurora. In addition, some RDS database types can use IAM roles instead of passwords for some use cases (for more detail, see Autentikasi basis data IAM).
OAuth tokens	Secret tokens – plain text string	Rotate: Store tokens in AWS Secrets Manager and configure automated rotation.
API tokens and keys	Secret tokens – plain text string	Rotate: Store in AWS Secrets Manager and establish automated rotation if possible.

Antipola umumnya adalah menyematkan kunci akses IAM ke dalam kode sumber, file konfigurasi, atau aplikasi seluler. Saat kunci akses IAM diperlukan untuk berkomunikasi dengan layanan AWS, gunakan kredensial keamanan sementara (jangka-pendek). Kredensial jangka pendek tersebut dapat diberikan melalui peran IAM untuk instans EC2, peran eksekusi untuk fungsi Lambda, peran IAM Cognito untuk akses pengguna seluler, dan kebijakan IoT Core untuk perangkat IoT. Saat beroperasi dengan pihak ketiga, utamakan mendelegasikan akses ke peran IAM dengan akses yang diperlukan ke sumber daya akun Anda daripada mengonfigurasikan pengguna IAM lalu mengirim kunci akses rahasia kepada pihak ketiga untuk pengguna tersebut.

Dalam banyak kasus, beban kerja memerlukan penyimpanan rahasia agar dapat saling beroperasi dengan sumber daya dan layanan lainnya. AWS Secrets Manager dibuat khusus untuk mengelola kredensial ini secara aman, sekaligus penyimpanan, penggunaan, dan rotasi token API, kata sandi, serta kredensial lainnya.

AWS Secrets Manager menyediakan lima kemampuan utama untuk memastikan keamanan penyimpanan dan penanganan kredensial sensitif: enkripsi diam, enkripsi bergerak, pengauditan menyeluruh, kontrol akses terperinci, dan rotasi kredensial yang dapat diperluas. Layanan manajemen rahasia lainnya dari Partner AWS atau solusi yang dikembangkan secara lokal yang memberikan kemampuan dan jaminan serupa juga dapat digunakan.

Langkah implementasi

1. Identifikasi jalur kode yang berisi kredensial hard-coding menggunakan alat otomatis seperti Amazon CodeGuru.
   - Gunakan Amazon CodeGuru untuk memindai repositori kode Anda. Setelah peninjauan selesai, filter `Type=Secrets` di CodeGuru untuk menemukan baris kode yang bermasalah.
2. Identifikasi kredensial yang dapat dihapus atau diganti.
   a. Identifikasi kredensial yang sudah tidak diperlukan, lalu tandai untuk dihapus.
   b. Untuk Kunci Rahasia AWS yang tersemat dalam kode sumber, ganti dengan peran IAM yang terkait dengan sumber daya yang diperlukan. Jika bagian beban kerja Anda berada di luar AWS tetapi memerlukan kredensial IAM untuk mengakses sumber daya AWS, pertimbangkan untuk menggunakan IAM Roles Anywhere atau AWS Systems Manager Hybrid Activations.
3. Untuk rahasia lama lainnya dari pihak ketiga yang memerlukan penggunaan strategi rotasi, integrasikan Secrets Manager ke dalam kode Anda untuk mengambil rahasia pihak ketiga pada waktu proses.
   a. Konsol CodeGuru dapat secara otomatis membuat rahasia di Secrets Manager menggunakan kredensial yang ditemukan.
   b. Integrasikan pengambilan rahasia dari Secrets Manager ke dalam kode aplikasi Anda.
      - Fungsi Lambda nirserver dapat menggunakan ekstensi Lambda bahasa agnostik.
      - Untuk instans atau kontainer EC2, AWS menyediakan contoh kode sisi klien untuk pengambilan rahasia dari Secrets Manager dalam beberapa bahasa pemrograman yang populer.
4. Tinjau basis kode Anda secara berkala dan pindai kembali untuk memverifikasi bahwa tidak ada rahasia baru yang ditambahkan ke kode.

- Pertimbangkan untuk menggunakan alat bantu seperti git-secrets untuk mencegah masuknya rahasia baru ke repositori kode sumber Anda.

5. Pantau aktivitas Secrets Manager untuk mengetahui apakah ada penggunaan yang tidak diharapkan, akses rahasia yang tidak sesuai, atau upaya penghapusan rahasia.
6. Kurangi akses manusia ke kredensial. Batasi akses membaca, menulis, dan memodifikasi kredensial untuk peran IAM khusus untuk tujuan ini, serta hanya sediakan akses untuk mengambil peran ke sebagian kecil pengguna operasional.

Sumber daya

Praktik Terbaik Terkait:

- SEC02-BP02 Menggunakan kredensial sementara
- SEC02-BP05 Mengaudit dan merotasi kredensial secara berkala

Dokumen terkait:

- Mulai menggunakan AWS Secrets Manager
- Penyedia Identitas dan Federasi
- Amazon CodeGuru Memperkenalkan Pendeteksi Rahasia
- Bagaimana AWS Secrets Manager menggunakan AWS Key Management Service
- Enkripsi dan dekripsi rahasia di Secrets Manager
- Entri blog Secrets Manager
- Amazon RDS mengumumkan integrasi dengan AWS Secrets Manager

Video terkait:

- Praktik Terbaik untuk Mengelola, Mengambil, dan Merotasi Rahasia dalam Skala Besar
- Temukan Rahasia yang Sudah Diberi Hard-Code Menggunakan Pendeteksi Rahasia Amazon CodeGuru
- Mengamankan Rahasia untuk Beban Kerja Hibrida Menggunakan AWS Secrets Manager

Lokakarya terkait:

- Menyimpan, mengambil, dan mengelola kredensial sensitif di AWS Secrets Manager
- AWS Systems Manager Hybrid Activations

SEC02-BP04 Mengandalkan penyedia identitas terpusat

Untuk identitas tenaga kerja (karyawan dan kontraktor), andalkan penyedia identitas yang memungkinkan Anda mengelola identitas di tempat terpusat. Ini akan mempermudah pengelolaan akses di beberapa aplikasi dan sistem, karena Anda membuat, menetapkan, mengelola, mencabut, dan mengaudit akses dari satu lokasi.

Hasil yang diinginkan: Anda memiliki penyedia identitas terpusat tempat Anda mengelola pengguna tenaga kerja, kebijakan autentikasi (misalnya mengharuskan autentikasi multifaktor (MFA)), dan otorisasi ke sistem dan aplikasi (misalnya menetapkan akses berdasarkan keanggotaan atau atribut grup pengguna) secara terpusat. Pengguna tenaga kerja Anda masuk ke penyedia identitas pusat dan melakukan penggabungan (masuk tunggal) ke aplikasi internal dan eksternal, sehingga pengguna tidak perlu mengingat lebih dari satu kredensial. Penyedia identitas Anda terintegrasi dengan sistem sumber daya manusia (SDM) Anda sehingga perubahan personel secara otomatis disinkronkan ke penyedia identitas Anda. Misalnya, jika seseorang keluar dari organisasi Anda, Anda dapat secara otomatis mencabut akses ke aplikasi dan sistem gabungan (termasuk AWS). Anda telah mengaktifkan pencatatan log audit mendetail di penyedia identitas Anda dan memantau log tersebut untuk perilaku pengguna yang tidak biasa.

Antipola umum:

- Anda tidak menggunakan federasi dan masuk tunggal. Pengguna tenaga kerja Anda membuat akun dan kredensial pengguna terpisah di beberapa aplikasi dan sistem.
- Anda belum mengotomatiskan siklus hidup identitas untuk pengguna tenaga kerja, seperti dengan mengintegrasikan penyedia identitas Anda dengan sistem SDM Anda. Saat pengguna keluar dari organisasi atau beralih jabatan, Anda mengikuti proses manual untuk menghapus atau memperbarui catatan mereka di beberapa aplikasi dan sistem.

Manfaat menjalankan praktik terbaik ini: Dengan menggunakan penyedia identitas yang terpusat, Anda memiliki satu tempat untuk mengelola identitas dan kebijakan pengguna tenaga kerja, kemampuan untuk menetapkan akses aplikasi kepada pengguna dan grup, dan kemampuan untuk memantau aktivitas masuk pengguna. Melalui integrasi dengan sistem sumber daya manusia (SDM), ketika pengguna beralih jabatan, perubahan ini disinkronkan dengan penyedia identitas yang secara otomatis memperbarui aplikasi dan izin yang ditetapkan. Ketika pengguna keluar dari organisasi

Anda, identitas mereka secara otomatis dinonaktifkan di penyedia identitas, sehingga akses mereka ke aplikasi dan sistem gabungan dicabut.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Panduan untuk pengguna tenaga kerja yang mengakses AWS

Pengguna tenaga kerja seperti karyawan dan kontraktor di organisasi Anda mungkin memerlukan akses ke AWS menggunakan AWS Management Console atau AWS Command Line Interface (AWS CLI) untuk menjalankan fungsi pekerjaan mereka. Anda dapat memberikan akses AWS kepada pengguna tenaga kerja Anda dengan melakukan federasi dari penyedia identitas terpusat Anda ke AWS pada dua tingkat: federasi langsung ke setiap Akun AWS atau federasi ke beberapa akun di organisasi AWS.

- Untuk menggabungkan pengguna tenaga kerja Anda secara langsung dengan setiap Akun AWS, Anda dapat menggunakan penyedia identitas terpusat untuk digabungkan ke AWS Identity and Access Management dalam akun tersebut. Fleksibilitas IAM memungkinkan Anda mengaktifkan Penyedia Identitas SAML 2.0 atau Open ID Connect (OIDC) secara terpisah untuk setiap Akun AWS dan gunakan atribut pengguna gabungan untuk kontrol akses. Pengguna tenaga kerja Anda akan menggunakan browser web mereka untuk masuk ke penyedia identitas dengan memberikan kredensialnya (seperti kata sandi dan kode token MFA). Penyedia identitas mengeluarkan pernyataan SAFL ke browser mereka yang dikirimkan ke URL masuk AWS Management Console agar pengguna dapat melakukan masuk tunggal ke AWS Management Console dengan mengambil Peran IAM. Pengguna Anda juga dapat memperoleh kredensial API AWS sementara untuk digunakan di AWS CLI atau SDK AWS dari AWS STS dengan mengambil peran IAM menggunakan pernyataan SAFL dari penyedia identitas.
- Untuk menggabungkan pengguna tenaga kerja Anda dengan beberapa akun di organisasi AWS Anda, Anda dapat menggunakan AWS IAM Identity Center untuk mengelola akses secara terpusat bagi pengguna tenaga kerja Anda ke Akun AWS dan aplikasi. Anda mengaktifkan Pusat Identitas untuk organisasi Anda dan mengonfigurasi sumber identitas Anda. IAM Identity Center menyediakan direktori sumber identitas default yang dapat Anda gunakan untuk mengelola pengguna dan grup Anda. Atau, Anda dapat memilih sumber identitas eksternal dengan terhubung ke penyedia identitas eksternal Anda menggunakan SAFL 2.0 dan secara otomatis menyediakan pengguna dan grup menggunakan SCIM, atau terhubung ke Direktori Microsoft AD Anda menggunakan AWS Directory Service. Setelah sumber identitas dikonfigurasi, Anda dapat menetapkan akses kepada pengguna dan grup ke Akun AWS dengan menentukan

kebijakan hak akses paling rendah di seperangkat izin. Pengguna tenaga kerja Anda dapat melakukan autentikasi melalui penyedia identitas pusat Anda untuk masuk ke portal akses AWS dan melakukan masuk tunggal ke Akun AWS dan aplikasi cloud yang ditetapkan untuk mereka. Pengguna Anda dapat mengonfigurasi AWS CLI v2 untuk mengautentikasi dengan Pusat Identitas dan mendapatkan kredensial untuk menjalankan perintah AWS CLI. Pusat Identitas juga memungkinkan akses masuk tunggal ke aplikasi AWS seperti Amazon SageMaker Studio dan portal AWS IoT Sitewise Monitor.

Setelah Anda mengikuti panduan di atas, pengguna tenaga kerja Anda tidak perlu lagi menggunakan IAM users dan grup IAM untuk operasi normal saat mengelola beban kerja di AWS. Sebaliknya, pengguna dan grup Anda dikelola di luar AWS dan pengguna dapat mengakses sumber daya AWS sebagai identitas gabungan. Identitas gabungan menggunakan grup yang ditentukan oleh penyedia identitas terpusat Anda. Anda harus mengidentifikasi dan menghapus grup IAM, IAM users, dan kredensial pengguna jangka panjang (kata sandi dan kunci akses) yang sudah tidak diperlukan di situs Akun AWS Anda. Anda dapat menemukan kredensial yang tidak digunakan menggunakan laporan kredensial IAM, menghapus IAM users terkait, dan menghapus grup IAM. Anda dapat menerapkan Kebijakan Kontrol Layanan (SCP) ke organisasi Anda yang membantu mencegah pembuatan IAM users dan grup IAM baru, sehingga memaksa akses ke AWS hanya terjadi melalui identitas gabungan.

Panduan untuk pengguna aplikasi Anda

Anda dapat mengelola identitas pengguna aplikasi Anda, seperti aplikasi seluler, menggunakan Amazon Cognito sebagai penyedia identitas terpusat Anda. Amazon Cognito memungkinkan autentikasi, otorisasi, dan manajemen pengguna untuk web dan aplikasi seluler Anda. Amazon Cognito menyediakan tempat penyimpanan identitas yang menyesuaikan skala dengan jutaan pengguna, mendukung federasi identitas sosial dan korporasi, serta menawarkan fitur keamanan canggih untuk membantu melindungi pengguna dan bisnis Anda. Anda dapat mengintegrasikan aplikasi web atau seluler kustom Anda dengan Amazon Cognito untuk menambahkan autentikasi pengguna dan kontrol akses ke aplikasi Anda dalam hitungan menit. Dibangun di atas standar identitas terbuka seperti SAFL dan Open ID Connect (OIDC), Amazon Cognito mendukung berbagai peraturan kepatuhan dan terintegrasi dengan sumber daya pengembangan frontend dan backend.

Langkah implementasi

Langkah-langkah untuk pengguna tenaga kerja yang mengakses AWS

- Gabungkan pengguna tenaga kerja Anda ke AWS menggunakan penyedia identitas terpusat melalui salah satu pendekatan berikut:
  - Gunakan IAM Identity Center untuk mengaktifkan masuk tunggal ke beberapa Akun AWS di organisasi AWS Anda dengan cara menggabungkan dengan penyedia identitas Anda.
  - Gunakan IAM untuk menghubungkan penyedia identitas Anda secara langsung ke setiap Akun AWS, sehingga memungkinkan akses mendetail gabungan.
- Identifikasikan dan hapus IAM users dan grup IAM yang digantikan dengan identitas gabungan.

Langkah-langkah untuk pengguna aplikasi Anda

- Gunakan Amazon Cognito sebagai penyedia identitas terpusat menuju aplikasi Anda.
- Integrasikan aplikasi kustom Anda dengan Amazon Cognito menggunakan OpenID Connect dan OAuth. Anda dapat mengembangkan aplikasi kustom menggunakan pustaka Amplify yang menyediakan antarmuka sederhana untuk diintegrasikan dengan berbagai layanan AWS, seperti Amazon Cognito untuk autentikasi.

Sumber daya

Praktik terbaik Well-Architected terkait:

- SEC02-BP06 Manfaatkan grup dan atribut pengguna
- SEC03-BP02 Memberikan hak akses paling rendah
- SEC03-BP06 Mengelola akses berdasarkan siklus hidup

Dokumen terkait:

- Federasi identitas di AWS
- Praktik terbaik keamanan di IAM
- Praktik terbaik AWS Identity and Access Management
- Memulai dengan administrasi terdelegasi IAM Identity Center
- Cara menggunakan kebijakan yang dikelola pelanggan di IAM Identity Center untuk kasus penggunaan lanjutan
- AWS CLI v2: penyedia kredensial IAM Identity Center

Video terkait:

- AWS re:Inforce 2022 - Pembahasan mendalam AWS Identity and Access Management (IAM)
- AWS re:invent 2022 - Menyederhanakan akses tenaga kerja Anda dengan IAM Identity Center
- AWS re:Invent 2018: Menguasai Identitas di Setiap Lapisan Susunan

Contoh terkait:

- Lokakarya: Menggunakan AWS IAM Identity Center untuk mencapai manajemen identitas yang kuat
- Lokakarya: Identitas nirserver

Alat terkait:

- Partner Kompetensi Keamanan AWS: Manajemen Identitas dan Akses
- saml2aws

SEC02-BP05 Mengaudit dan merotasi kredensial secara berkala

Audit dan rotasikan kredensial secara berkala guna membatasi seberapa lama kredensial dapat digunakan untuk mengakses sumber daya Anda. Kredensial jangka panjang menimbulkan banyak risiko, tetapi risiko ini dapat dikurangi dengan merotasikan kredensial jangka panjang secara berkala.

Hasil yang diinginkan: Mengimplementasikan rotasi kredensial untuk mengurangi risiko terkait penggunaan kredensial jangka panjang. Melakukan audit dan perbaikan secara rutin untuk penggunaan yang tidak mematuhi kebijakan rotasi kredensial.

Antipola umum:

- Tidak mengaudit penggunaan kredensial.
- Menggunakan kredensial jangka panjang saat tidak diperlukan.
- Menggunakan kredensial jangka panjang dan tidak merotasinya secara rutin.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Jika Anda tidak dapat mengandalkan kredensial sementara dan memerlukan kredensial jangka panjang, lakukan audit kredensial untuk memastikan bahwa kontrol yang ditentukan seperti autentikasi multi-faktor (MFA) telah diterapkan, dirotasi secara rutin, dan memiliki tingkat akses yang sesuai.

Validasi berkala, sebaiknya melalui alat otomatis, diperlukan untuk memverifikasi penerapan kontrol yang tepat. Untuk identitas manusia, Anda harus mewajibkan pengguna untuk mengubah kata sandi mereka secara rutin dan menonaktifkan kunci akses yang ditukar dengan kredensial sementara. Setelah Anda beralih dari pengguna AWS Identity and Access Management (IAM) ke pengguna identitas terpusat, Anda dapat membuat laporan kredensial untuk mengaudit pengguna Anda.

Anda juga sebaiknya menerapkan dan memantau MFA dalam penyedia identitas Anda. Anda dapat mengonfigurasikan Aturan AWS Config, atau menggunakan Standar Keamanan AWS Security Hub, untuk memantau apakah pengguna mengaktifkan MFA. Pertimbangkan untuk menggunakan IAM Roles Anywhere guna memberikan kredensial sementara untuk identitas mesin. Dalam situasi yang tidak memungkinkan penggunaan peran IAM dan kredensial sementara, pengauditan dan rotasi kunci akses perlu sering dilakukan.

Langkah implementasi

- Audit kredensial secara rutin: Mengaudit identitas yang dikonfigurasikan dalam penyedia identitas dan IAM Anda membantu memastikan bahwa hanya identitas yang diotorisasi yang memiliki akses ke beban kerja Anda. Identitas tersebut mencakup, tetapi tidak terbatas pada, pengguna IAM, pengguna AWS IAM Identity Center, pengguna Active Directory, atau pengguna dalam penyedia identitas hulu yang berbeda. Misalnya, hapus orang yang keluar dari organisasi, serta hapus peran lintas akun yang sudah tidak diperlukan. Terapkan proses untuk secara berkala mengaudit izin ke layanan yang diakses oleh entitas IAM. Tindakan ini akan membantu Anda mengidentifikasi kebijakan yang perlu diubah untuk menghapus izin yang tidak digunakan. Gunakan laporan kredensial dan AWS Identity and Access Management Access Analyzer untuk mengaudit izin dan kredensial IAM. Anda dapat menggunakan Amazon CloudWatch untuk mengonfigurasi alarm untuk panggilan API tertentu dalam lingkungan AWS Anda. Amazon GuardDuty juga dapat memberikan peringatan terkait aktivitas yang tidak diharapkan, yang mungkin menandakan akses yang terlalu permisif atau akses yang tidak diinginkan ke kredensial IAM.
- Lakukan rotasi kredensial secara rutin: Ketika Anda tidak dapat menggunakan kredensial sementara, rotasikan kunci akses IAM jangka panjang secara rutin (maksimum 90 hari sekali). Tindakan ini akan membatasi waktu penggunaan kredensial untuk mengakses sumber daya Anda

jika kunci akses bocor tanpa sepengetahuan Anda. Untuk informasi tentang merotasi kunci akses bagi pengguna IAM, lihat Merotasi kunci akses.

- Tinjau izin IAM: Untuk meningkatkan keamanan Akun AWS Anda, tinjau dan pantau setiap kebijakan IAM Anda secara rutin. Pastikan kebijakan tersebut memenuhi prinsip hak akses paling rendah.
- Pertimbangkan untuk mengotomatiskan pembaruan dan pembuatan sumber daya IAM: IAM Identity Center mengotomatiskan banyak tugas IAM, seperti manajemen kebijakan dan peran. Atau, AWS CloudFormation dapat digunakan untuk mengotomatiskan deployment sumber daya IAM, termasuk kebijakan dan peran, untuk mengurangi kemungkinan kesalahan akibat kelalaian manusia karena templat dapat diverifikasi serta dikelola dengan kendali versi.
- Gunakan IAM Roles Anywhere untuk mengganti pengguna IAM untuk identitas mesin: IAM Roles Anywhere memungkinkan Anda untuk menggunakan peran dalam area yang secara tradisional tidak bisa digunakan, seperti server on-premise. IAM Roles Anywhere menggunakan sertifikat X.509 tepercaya untuk mengautentikasi ke AWS serta menerima kredensial sementara. Dengan IAM Roles Anywhere, Anda tidak perlu merotasi kredensial ini karena kredensial jangka panjang tidak lagi disimpan dalam lingkungan on-premise Anda. Perlu diketahui bahwa Anda harus memantau dan merotasi sertifikat X.509 sebelum kedaluwarsa.

Sumber daya

Praktik Terbaik Terkait:

- SEC02-BP02 Menggunakan kredensial sementara
- SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman

Dokumen terkait:

- Mulai menggunakan AWS Secrets Manager
- Praktik Terbaik IAM
- Penyedia Identitas dan Federasi
- Solusi Partner Keamanan: Akses dan Kontrol Akses
- Kredensial Keamanan Sementara
- Mendapatkan laporan kredensial untuk Akun AWS Anda

Video terkait:

- Praktik Terbaik untuk Mengelola, Mengambil, dan Merotasi Rahasia dalam Skala Besar
- Mengelola izin pengguna dalam skala besar dengan AWS IAM Identity Center
- Menguasai identitas di setiap lapisan susunan

Contoh terkait:

- Lab Well-Architected - Pembersihan Pengguna IAM Otomatis
- Lab Well-Architected Deployment Otomatis Peran dan Grup IAM

SEC02-BP06 Manfaatkan grup dan atribut pengguna

Seiring meningkatnya jumlah pengguna yang dikelola, Anda perlu menentukan cara agar dapat mengelolanya dalam skala besar. Tempatkan pengguna yang memiliki persyaratan keamanan yang sama dalam grup yang ditentukan oleh penyedia identitas Anda, dan terapkan mekanisme untuk memastikan atribut pengguna yang dapat digunakan untuk kontrol akses (misalnya departemen atau lokasi) sudah benar dan diperbarui. Gunakan grup dan atribut tersebut untuk mengontrol akses, bukan pengguna individual. Dengan demikian, Anda dapat mengelola akses secara terpusat cukup dengan satu kali mengubah keanggotaan atau atribut grup pengguna dengan seperangkat izin, daripada memperbarui banyak kebijakan satu per satu saat akses pengguna perlu diubah. Anda dapat menggunakan AWS IAM Identity Center (IAM Identity Center) untuk mengelola grup dan atribut pengguna. IAM Identity Center mendukung atribut yang paling sering digunakan, baik dimasukkan secara manual selama pembuatan pengguna atau disediakan secara otomatis menggunakan mesin sinkronisasi, seperti yang ditetapkan dalam spesifikasi Sistem untuk Manajemen Identitas Lintas Domain (SCIM).

Tempatkan pengguna yang memiliki persyaratan keamanan yang sama dalam grup yang ditentukan oleh penyedia identitas Anda, dan terapkan mekanisme untuk memastikan atribut pengguna yang dapat digunakan untuk kontrol akses (misalnya departemen atau lokasi) sudah benar dan diperbarui. Gunakan grup dan atribut tersebut, bukan pengguna individual, untuk mengontrol akses. Dengan demikian, Anda dapat mengelola akses secara terpusat cukup dengan satu kali mengubah keanggotaan atau atribut grup pengguna, daripada memperbarui banyak kebijakan satu per satu saat akses pengguna perlu diubah.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Jika Anda menggunakan AWS IAM Identity Center (IAM Identity Center), konfigurasikan grup: IAM Identity Center memberikan kemampuan untuk mengonfigurasi grup pengguna dan menetapkan grup untuk tingkat izin yang diinginkan.
  - AWS Masuk Tunggal - Kelola Identitas
- Pelajari lebih lanjut tentang kontrol akses berbasis atribut (ABAC): ABAC adalah strategi otorisasi yang menetapkan izin berdasarkan atribut.
  - Apa Itu ABAC untuk AWS?
  - Lab: Kontrol Akses Berbasis Tanda IAM untuk EC2

Sumber daya

Dokumen terkait:

- Mulai Menggunakan AWS Secrets Manager
- Praktik Terbaik IAM
- Penyedia Identitas dan Federasi
- Pengguna Root Akun AWS

Video terkait:

- Praktik Terbaik untuk Mengelola, Mengambil, dan Merotasi Secret dalam Skala Besar
- Mengelola izin pengguna dalam skala besar dengan AWS IAM Identity Center
- Menguasai identitas di setiap lapisan beban kerja

Contoh terkait:

- Lab: Kontrol Akses Berbasis Tanda IAM untuk EC2

Manajemen izin

Kelola izin guna mengontrol akses untuk identitas orang dan mesin yang memerlukan akses ke AWS dan beban kerja Anda. Izin mengontrol cakupan dan ketentuan akses seseorang. Tetapkan

izin kepada identitas manusia dan mesin tertentu guna memberikan akses ke tindakan layanan tertentu atas sumber daya tertentu. Selain itu, tentukan syarat yang harus dipenuhi agar akses dapat diberikan. Sebagai contoh, Anda dapat mengizinkan developer membuat fungsi Lambda baru, tetapi hanya di Wilayah tertentu. Saat mengelola lingkungan AWS Anda dalam skala yang besar, ikuti praktik terbaik berikut guna memastikan bahwa identitas hanya diberikan akses yang diperlukan, tidak lebih dari itu.

Terdapat beberapa cara untuk memberikan akses ke beberapa jenis sumber daya yang berbeda. Salah satunya adalah menggunakan beberapa jenis kebijakan yang berbeda.

Kebijakan berbasis identitas di IAM dikelola atau sebaris dan dilampirkan ke identitas IAM, termasuk pengguna, grup, atau aturan. Dengan kebijakan ini, Anda dapat menentukan apa saja yang boleh dilakukan identitas tersebut (izinnya). Kebijakan berbasis identitas dapat dikategorikan lebih lanjut.

Kebijakan terkelola – Kebijakan berbasis identitas tersendiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di akun AWS Anda. Terdapat dua jenis kebijakan terkelola:

- Kebijakan terkelola AWS – Kebijakan terkelola yang dibuat dan dikelola oleh AWS.
- Kebijakan yang dikelola pelanggan – Kebijakan terkelola yang Anda buat dan kelola di akun AWS Anda. Kebijakan yang dikelola pelanggan memberikan kontrol yang lebih presisi terhadap kebijakan Anda dibandingkan dengan kebijakan yang dikelola AWS.

Kebijakan terkelola adalah metode yang diutamakan untuk menerapkan izin. Namun, Anda juga dapat menggunakan kebijakan sebaris yang Anda tambahkan langsung ke pengguna, grup, atau peran tunggal. Kebijakan sebaris menjaga hubungan satu-ke-satu antara kebijakan dan identitas. Kebijakan sebaris akan dihapus saat Anda menghapus identitas.

Di sebagian besar kasus, Anda harus membuat sendiri kebijakan yang dikelola pelanggan dengan mengikuti prinsip hak akses paling rendah.

Kebijakan berbasis sumber daya dilampirkan ke sumber daya. Sebagai contoh, kebijakan bucket S3 merupakan kebijakan berbasis sumber daya. Kebijakan ini memberikan izin kepada pengguna utama yang dapat berada di akun yang sama atau berbeda dengan sumber daya. Untuk daftar layanan yang mendukung kebijakan berbasis sumber daya, lihat layanan AWS yang dapat digunakan dengan IAM.

Batasan izin menggunakan kebijakan terkelola untuk menetapkan izin maksimum yang dapat ditetapkan administrator. Dengan demikian, Anda dapat mendelegasikan kemampuan untuk

membuat dan mengelola izin kepada developer, seperti pembuatan peran IAM, tetapi membatasi izin yang dapat mereka berikan agar mereka tidak dapat memperluas izin mereka menggunakan izin yang telah mereka buat.

Kontrol akses berbasis atribut (ABAC) memungkinkan Anda memberikan izin berdasarkan atribut. Di AWS, ini disebut sebagai tanda. Tanda dapat dilampirkan pada pengguna utama IAM (pengguna atau peran) dan pada sumber daya AWS. Dengan kebijakan IAM, administrator dapat membuat kebijakan yang dapat digunakan kembali yang menerapkan izin berdasarkan atribut pengguna utama IAM. Sebagai contoh, sebagai administrator Anda dapat menggunakan kebijakan IAM tunggal yang memberi developer di organisasi Anda akses ke sumber daya AWS yang cocok dengan tanda proyek developer. Seiring tim developer menambahkan sumber daya ke proyek, izin diterapkan secara otomatis berdasarkan atribut. Dengan demikian, tidak diperlukan pembaruan kebijakan untuk setiap sumber daya baru.

Kebijakan kontrol layanan (SCP) organisasi menetapkan izin maksimum untuk anggota akun organisasi atau unit organisasi (OU). SCP membatasi izin yang diberikan oleh kebijakan berbasis identitas atau kebijakan berbasis sumber daya kepada entitas (pengguna atau peran) dalam akun tersebut, tetapi tidak memberikan izin.

Kebijakan sesi mengasumsikan peran atau pengguna gabungan. Lewati kebijakan sesi saat menggunakan kebijakan Sesi CLI AWS atau API AWS untuk membatasi izin yang diberikan oleh kebijakan berbasis identitas peran atau pengguna ke sesi tersebut. Kebijakan ini membatasi izin untuk sesi yang dibuat, tetapi tidak memberikan izin. Untuk informasi selengkapnya, lihat Kebijakan Sesi.

Praktik terbaik

- SEC03-BP01 Menetapkan persyaratan akses
- SEC03-BP02 Memberikan hak akses paling rendah
- SEC03-BP03 Menerapkan proses akses darurat
- SEC03-BP04 Mengurangi izin secara terus-menerus
- SEC03-BP05 Menentukan pagar pembatas izin untuk organisasi Anda
- SEC03-BP06 Mengelola akses berdasarkan siklus hidup
- SEC03-BP07 Menganalisis akses lintas akun dan publik
- SEC03-BP08 Membagikan sumber daya secara aman dalam organisasi Anda
- SEC03-BP09 Membagikan sumber daya secara aman kepada pihak ketiga

SEC03-BP01 Menetapkan persyaratan akses

Tiap-tiap komponen atau sumber daya beban kerja Anda perlu diakses oleh administrator, pengguna akhir, atau komponen lainnya. Miliki penetapan yang jelas tentang siapa atau apa yang harus memiliki akses ke tiap-tiap komponen, pilih tipe identitas dan metode autentikasi serta otorisasi yang tepat.

Antipola umum:

- Hard-coding atau menyimpan rahasia di dalam aplikasi Anda.
- Memberikan izin kustom untuk tiap pengguna.
- Menggunakan kredensial berumur panjang.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Tiap-tiap komponen atau sumber daya beban kerja Anda perlu diakses oleh administrator, pengguna akhir, atau komponen lainnya. Miliki penetapan yang jelas tentang siapa atau apa yang harus memiliki akses ke tiap-tiap komponen, pilih tipe identitas dan metode autentikasi serta otorisasi yang tepat.

Akses rutin ke Akun AWS di dalam organisasi harus disediakan menggunakan akses gabungan atau penyedia identitas terpusat. Anda juga sebaiknya memusatkan manajemen identitas Anda dan memastikan terdapat praktik yang matang untuk mengintegrasikan akses AWS ke siklus hidup akses karyawan Anda. Misalnya, saat seorang karyawan berganti peran pekerjaan dengan level akses berbeda, keanggotaan grupnya juga harus berubah agar sesuai dengan persyaratan akses barunya.

Saat menetapkan persyaratan akses untuk identitas non-manusia, tentukan aplikasi dan komponen mana yang memerlukan akses dan bagaimana izin diberikan. Menggunakan IAM role yang dibangun dengan model akses hak akses paling rendah adalah pendekatan yang disarankan. Kebijakan yang Dikelola AWS menyediakan kebijakan IAM yang telah ditetapkan sebelumnya yang mencakup kasus-kasus penggunaan paling umum.

Layanan AWS, seperti AWS Secrets Manager dan AWS Systems Manager Parameter Store, dan membantu memisahkan rahasia dari aplikasi atau beban kerja secara aman pada kasus-kasus yang tidak memungkinkan penggunaan IAM role. Di Secrets Manager, Anda dapat membuat rotasi otomatis untuk kredensial Anda. Anda dapat menggunakan Systems Manager untuk

merujuk parameter di skrip, perintah, dokumen SSM, konfigurasi, dan alur kerja otomatisasi Anda menggunakan nama unik yang telah Anda tentukan saat membuat parameter tersebut.

Anda dapat menggunakan AWS Identity and Access Management Roles Anywhere untuk mendapatkan kredensial keamanan sementara di IAM untuk beban kerja yang berjalan di luar AWS. Beban kerja Anda dapat menggunakan kebijakan IAM dan IAM role yang sama dengan yang Anda gunakan dengan aplikasi AWS untuk mengakses sumber daya AWS.

Jika memungkinkan, gunakan kredensial sementara jangka pendek, bukan kredensial statis jangka panjang. Untuk skenario di mana Anda memerlukan pengguna IAM dengan akses terprogram dan kredensial jangka panjang, gunakan informasi yang terakhir digunakan kunci akses untuk merotasi dan menghapus kunci akses.

Sumber daya

Dokumen terkait:

- Kontrol akses berbasis atribut (ABAC)
- AWS IAM Identity Center
- IAM Roles Anywhere
- Kebijakan yang dikelola AWS untuk IAM Identity Center
- Ketentuan kebijakan AWS IAM
- Kasus penggunaan IAM
- Hapus kredensial yang tidak diperlukan
- Bekerja dengan Kebijakan
- Cara mengontrol akses ke sumber daya AWS berdasarkan Akun AWS, OU, atau organisasi
- Identifikasi, atur, dan kelola rahasia secara mudah menggunakan pencarian yang ditingkatkan di AWS Secrets Manager

Video terkait:

- Menjadi Master Kebijakan IAM dalam 60 Menit atau Kurang
- Pemisahan Tugas, Hak Akses Paling Rendah, Delegasi, dan CI/CD
- Merampingkan manajemen identitas dan akses untuk inovasi

SEC03-BP02 Memberikan hak akses paling rendah

Salah satu praktik terbaik adalah memberikan hanya akses yang diperlukan identitas untuk melakukan tindakan tertentu pada sumber daya tertentu dalam kondisi tertentu. Gunakan atribut grup dan identitas untuk menetapkan izin secara dinamis dalam skala besar daripada menentukan izin satu per satu untuk setiap pengguna. Misalnya, Anda dapat memberikan akses kepada sebuah grup developer untuk mengelola sumber daya untuk proyek mereka saja. Dengan cara ini, jika seorang developer keluar dari proyek, akses developer tersebut secara otomatis dicabut tanpa mengubah kebijakan akses dasar.

Hasil yang diinginkan: Pengguna hanya memiliki izin yang diperlukan untuk melakukan pekerjaannya. Pengguna hanya diberi akses ke lingkungan produksi untuk melakukan tugas tertentu dalam jangka waktu terbatas dan akses harus dicabut setelah tugas tersebut selesai. Izin harus dicabut jika sudah tidak digunakan, termasuk saat pengguna beralih ke proyek atau jabatan kerja lain. Hak akses administrator hanya boleh diberikan kepada sekelompok kecil administrator yang tepercaya. Izin harus ditinjau secara rutin untuk menghindari creep izin. Akun sistem atau mesin hanya boleh diberi rangkaian izin paling sedikit yang diperlukan untuk menyelesaikan tugas mereka.

Antipola umum:

- Memberikan izin administrator kepada para pengguna secara default.
- Menggunakan pengguna root untuk aktivitas harian.
- Membuat kebijakan yang terlalu permisif, tetapi tanpa hak istimewa administrator penuh.
- Tidak meninjau izin untuk mengetahui apakah izin tersebut memberikan hak akses paling rendah.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Prinsip hak akses paling rendah menyatakan bahwa identitas hanya boleh diizinkan untuk melakukan rangkaian tindakan sekecil mungkin yang diperlukan untuk menyelesaikan tugas tertentu. Hal ini menyeimbangkan kegunaan, efisiensi, dan keamanan. Pengoperasian berdasarkan prinsip ini membantu membatasi akses yang tidak diinginkan dan membantu memantau siapa saja yang memiliki akses ke sumber daya yang mana. Pengguna dan peran IAM secara default tidak memiliki izin apa pun. Pengguna root memiliki akses penuh secara default dan harus secara ketat dikontrol, dimonitor, dan digunakan hanya untuk tugas yang memerlukan akses root.

Kebijakan IAM digunakan untuk memberikan izin secara eksplisit ke peran IAM atau sumber daya tertentu. Contohnya, kebijakan berbasis identitas dapat dilampirkan ke grup IAM, sedangkan bucket S3 dapat dikontrol oleh kebijakan berbasis sumber daya.

Saat membuat kebijakan IAM, Anda dapat menentukan tindakan layanan, sumber daya, dan kondisi yang harus terpenuhi agar AWS dapat memberikan atau menolak akses. AWS mendukung beragam kondisi untuk membantu Anda menyaring akses. Contohnya, dengan menggunakan kunci kondisi `PrincipalOrgID`, Anda dapat menolak tindakan jika pemohon bukan bagian dari Organisasi AWS Anda.

Anda juga dapat mengontrol permintaan yang dibuat oleh layanan AWS atas nama Anda, seperti AWS CloudFormation yang membuat fungsi AWS Lambda, dengan menggunakan kunci kondisi `CalledVia`. Anda sebaiknya menggunakan berbagai macam kebijakan secara berlapis untuk membuat sistem pertahanan yang mendalam dan membatasi izin keseluruhan untuk pengguna Anda. Anda juga bisa membatasi izin yang dapat diberikan beserta kondisinya. Misalnya, Anda dapat mengizinkan tim aplikasi membuat kebijakan IAM sendiri untuk sistem yang mereka buat, tetapi Anda juga harus menerapkan Batasan Izin untuk membatasi izin maksimum yang bisa didapatkan sistem.

Langkah implementasi

- Implementasikan kebijakan hak akses paling rendah: Tetapkan kebijakan akses dengan hak paling rendah ke grup dan peran IAM untuk mencerminkan peran atau fungsi pengguna yang telah Anda tetapkan.
  - Kebijakan dasar untuk penggunaan API: Salah satu cara untuk menentukan izin yang diperlukan adalah dengan meninjau log AWS CloudTrail. Dengan peninjauan ini, Anda dapat membuat izin yang disesuaikan dengan tindakan yang benar-benar dilakukan oleh pengguna di dalam AWS. IAM Access Analyzer dapat menghasilkan kebijakan IAM secara otomatis berdasarkan aktivitas. Anda dapat menggunakan Penasihat Akses IAM di tingkat organisasi atau akun guna melacak informasi yang terakhir diakses untuk kebijakan tertentu.
- Pertimbangkan penggunaan kebijakan terkelola AWS untuk fungsi tugas. Saat akan membuat kebijakan izin yang disesuaikan secara mendetail, mungkin sulit untuk mengetahui cara memulainya. AWS memiliki kebijakan terkelola untuk peran tugas umum, misalnya penagihan, administrator basis data, dan ilmuwan data. Kebijakan ini dapat membantu mempersempit akses yang dimiliki pengguna selagi menentukan cara menerapkan kebijakan hak akses paling rendah.
- Hapus izin yang tidak diperlukan: Hapus izin yang tidak diperlukan dan ketatkan kebijakan yang terlalu longgar. Pembuatan kebijakan IAM Access Analyzer dapat membantu menyaring kebijakan izin.

- Pastikan pengguna memiliki akses yang terbatas ke lingkungan produksi: Pengguna seharusnya hanya memiliki akses ke lingkungan produksi dengan kasus penggunaan yang valid. Setelah pengguna menyelesaikan tugas tertentu yang memerlukan akses produksi, akses harus dicabut. Pembatasan akses ke lingkungan produksi membantu mencegah kejadian tak terduga yang memengaruhi produksi dan memperkecil cakupan dampak akses yang tidak diharapkan.
- Pertimbangkan batasan izin: Batasan izin adalah fitur untuk menggunakan kebijakan terkelola yang menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM. Batasan izin memungkinkan entitas melakukan tindakan hanya yang diizinkan oleh kebijakan berbasis identitas serta batasan izinnya.
- Pertimbangkan tanda sumber daya untuk izin: Model kontrol akses berbasis atribut yang menggunakan tanda sumber daya memungkinkan Anda memberikan akses berdasarkan tujuan sumber daya, pemilik, lingkungan, atau kriteria lain. Misalnya, Anda dapat menggunakan tanda sumber daya untuk membedakan lingkungan pengembangan dan produksi. Dengan tanda ini, Anda dapat membatasi developer agar hanya dapat mengakses lingkungan pengembangan. Dengan memadukan kebijakan pemberian tanda dan izin, Anda dapat memiliki akses sumber daya yang terperinci tanpa harus menentukan kebijakan kustom dan rumit untuk setiap fungsi tugas.
- Gunakan kebijakan kontrol layanan untuk AWS Organizations. Kebijakan kontrol layanan secara terpusat mengontrol izin maksimum yang tersedia bagi akun anggota di organisasi Anda. Fungsi penting dari kebijakan kontrol layanan adalah memungkinkan Anda membatasi izin pengguna root di dalam akun anggota. Pertimbangkan juga untuk menggunakan AWS Control Tower, yang menyediakan kontrol terkelola preskriptif yang makin melengkapi AWS Organizations. Anda juga dapat menentukan kontrol sendiri di dalam Control Tower.
- Buat kebijakan siklus hidup pengguna untuk organisasi Anda: Kebijakan siklus hidup pengguna menentukan tugas yang harus dilakukan saat pengguna ditambahkan ke AWS, mengubah peran atau cakupan pekerjaan, atau sudah tidak memerlukan akses ke AWS. Peninjauan izin harus dilakukan selama setiap langkah dalam siklus hidup pengguna untuk memverifikasi bahwa izin dibatasi dengan sesuai dan untuk menghindari creep izin.
- Buat jadwal rutin untuk meninjau izin dan menghapus izin yang tidak diperlukan: Anda harus rutin meninjau akses pengguna untuk memverifikasi bahwa pengguna tidak memiliki akses yang terlalu leluasa. AWS Config dan IAM Access Analyzer dapat membantu Anda saat mengaudit izin pengguna.
- Buat matriks peran kerja: Matrik peran kerja memberikan visualiasi untuk berbagai peran dan tingkat akses yang diperlukan dalam jejak AWS Anda. Dengan matriks peran kerja, Anda dapat menentukan dan memisahkan izin berdasarkan tanggung jawab pengguna di dalam organisasi. Gunakan grup daripada menerapkan izin secara langsung ke pengguna atau peran satu per satu.

Sumber Daya

Dokumen terkait:

- Berikan hak akses paling rendah
- Batasan izin untuk entitas IAM
- Teknik untuk menulis kebijakan IAM hak akses paling rendah
- IAM Access Analyzer mempermudah implementasi izin hak akses paling rendah dengan menghasilkan kebijakan IAM berdasarkan aktivitas akses
- Delegasikan manajemen izin ke developer menggunakan batasan izin IAM
- Mempersempit Izin menggunakan informasi yang terakhir kali diakses
- Jenis kebijakan IAM dan kapan harus digunakan
- Menguji kebijakan IAM dengan simulator kebijakan IAM
- Pagar Pembatas di AWS Control Tower
- Arsitektur Zero Trust: Sebuah perspektif AWS
- Cara mengimplementasikan prinsip hak akses paling rendah dengan CloudFormation StackSets
- Kontrol akses berbasis atribut (ABAC)
- Mengurangi cakupan kebijakan dengan melihat aktivitas pengguna
- Lihat akses peran
- Gunakan Penandaan untuk Mengelola Lingkungan Anda dan Meningkatkan Akuntabilitas
- Strategi Penandaan AWS
- Penandaan sumber daya AWS

Video terkait:

- Manajemen izin generasi baru
- Zero Trust: Sebuah perspektif AWS
- Bagaimana cara menggunakan batasan izin untuk membatasi pengguna dan peran guna mencegah eskalasi hak akses?

Contoh terkait:

- Lab: Batasan izin IAM yang mendelegasikan pembuatan peran

- Lab: Kontrol akses berbasis tanda IAM untuk EC2

SEC03-BP03 Menerapkan proses akses darurat

Buat proses yang memungkinkan akses darurat ke beban kerja Anda jika terjadi masalah pada penyedia identitas terpusat Anda.

Anda harus merancang proses untuk berbagai mode kegagalan yang dapat mengakibatkan peristiwa darurat. Misalnya, dalam keadaan normal, pengguna tenaga kerja Anda melakukan federasi ke cloud menggunakan penyedia identitas terpusat (SEC02-BP04) untuk mengelola beban kerja mereka. Namun, jika penyedia identitas terpusat Anda gagal, atau konfigurasi untuk federasi di cloud diubah, maka pengguna tenaga kerja Anda mungkin tidak dapat melakukan federasi ke cloud. Proses akses darurat memungkinkan administrator yang berwenang untuk mengakses sumber daya cloud Anda melalui cara alternatif (seperti bentuk federasi alternatif atau akses pengguna langsung) untuk memperbaiki masalah dengan konfigurasi federasi atau beban kerja Anda. Proses akses darurat digunakan sampai mekanisme federasi normal dipulihkan.

Hasil yang diinginkan:

- Anda telah menentukan dan mendokumentasikan mode kegagalan yang terhitung sebagai keadaan darurat: pertimbangkan keadaan normal Anda dan sistem yang diandalkan oleh pengguna Anda untuk mengelola beban kerja mereka. Pertimbangkan bagaimana setiap dependensi ini dapat gagal dan menyebabkan keadaan darurat. Anda dapat menemukan pertanyaan dan praktik terbaik di Pilar Keandalan yang berguna untuk mengidentifikasi mode kegagalan dan merancang sistem yang lebih tangguh untuk meminimalkan kemungkinan kegagalan.
- Anda telah mendokumentasikan langkah-langkah yang harus diikuti untuk mengonfirmasi kegagalan sebagai keadaan darurat. Misalnya, Anda dapat meminta administrator identitas Anda untuk memeriksa status penyedia identitas utama dan siaga Anda dan, jika keduanya tidak tersedia, umumkan peristiwa darurat untuk kegagalan penyedia identitas.
- Anda telah menentukan proses akses darurat khusus untuk setiap jenis mode darurat atau kegagalan. Pengkhususan ini dapat mengurangi godaan di pihak pengguna Anda untuk terlalu sering menggunakan proses umum untuk semua jenis keadaan darurat. Proses akses darurat Anda menggambarkan keadaan di mana setiap proses harus digunakan dan, sebaliknya, situasi di mana proses tidak boleh digunakan dan menunjuk ke proses alternatif yang mungkin berlaku.
- Proses Anda didokumentasikan dengan baik dengan instruksi yang mendetail dan playbook yang dapat diikuti dengan cepat dan efisien. Ingatlah bahwa peristiwa darurat dapat menjadi saat yang

memusingkan bagi pengguna Anda dan mereka sedang di bawah tekanan waktu yang ekstrem, jadi rancanglah proses Anda sesederhana mungkin.

Antipola umum:

- Anda tidak memiliki proses akses darurat yang terdokumentasi dengan baik dan teruji dengan baik. Pengguna Anda tidak siap menghadapi keadaan darurat dan mengikuti proses improvisasi ketika peristiwa darurat muncul.
- Proses akses darurat Anda bergantung pada sistem yang sama (seperti penyedia identitas terpusat) dengan mekanisme akses normal Anda. Ini artinya, kegagalan sistem tersebut dapat memengaruhi mekanisme akses normal dan darurat Anda dan mengganggu kemampuan Anda untuk pulih dari kegagalan.
- Proses akses darurat Anda digunakan dalam situasi non-darurat. Misalnya, pengguna Anda sering menyalahgunakan proses akses darurat karena mereka merasa lebih mudah melakukan perubahan secara langsung daripada mengirimkan perubahan melalui pipeline.
- Proses akses darurat Anda tidak menghasilkan log yang memadai untuk mengaudit proses, atau log tersebut tidak dipantau untuk mendapatkan peringatan potensi penyalahgunaan proses.

Manfaat menjalankan praktik terbaik ini:

- Dengan memiliki proses akses darurat yang terdokumentasi dengan baik dan teruji dengan baik, Anda dapat mengurangi waktu yang dibutuhkan pengguna untuk merespons dan menyelesaikan peristiwa darurat. Hal ini dapat menghasilkan lebih sedikit waktu henti dan ketersediaan yang lebih tinggi untuk layanan yang Anda berikan kepada pelanggan Anda.
- Anda dapat melacak setiap permintaan akses darurat dan mendeteksi serta memberikan peringatan adanya upaya penyalahgunaan proses untuk peristiwa non-darurat.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Bagian ini memberikan panduan dalam membuat proses akses darurat untuk beberapa mode kegagalan yang berkaitan dengan beban kerja yang di-deploy di AWS, dimulai dengan panduan umum yang berlaku untuk semua mode kegagalan dan dilanjutkan dengan panduan khusus berdasarkan jenis mode kegagalan.

Panduan umum untuk semua mode kegagalan

Pertimbangkan hal berikut saat Anda merancang proses akses darurat untuk mode kegagalan:

- Dokumentasikan kondisi awal dan asumsi tentang proses tersebut: kapan proses tersebut harus digunakan dan kapan proses tersebut tidak boleh digunakan. Penting untuk memiliki detail mode kegagalan dan mendokumentasikan asumsi, seperti keadaan sistem terkait lainnya. Misalnya, proses untuk Mode Kegagalan 2 mengasumsikan bahwa penyedia identitas tersedia, tetapi konfigurasi di AWS dimodifikasi atau telah kedaluwarsa.
- Sejak awal, buat sumber daya yang dibutuhkan oleh proses akses darurat (SEC10-BP05). Misalnya, buat Akun AWS akses darurat di awal dengan IAM users dan peran IAM, dan peran IAM lintas akun di semua akun beban kerja. Hal ini memastikan bahwa semua sumber daya ini siap dan tersedia ketika peristiwa darurat terjadi. Dengan membuat sumber daya di awal, Anda tidak bergantung pada API AWS bidang kendali (yang digunakan untuk membuat dan memodifikasi sumber daya AWS) yang mungkin tidak tersedia dalam keadaan darurat. Selanjutnya, dengan membuat sumber daya IAM di awal, Anda tidak perlu memperhitungkan potensi penundaan disebabkan konsistensi akhir.
- Sertakan proses akses darurat sebagai bagian dari rencana manajemen insiden Anda (SEC10-BP02). Dokumentasikan bagaimana peristiwa darurat dilacak dan dikomunikasikan kepada orang lain di organisasi Anda seperti tim sejawat, pimpinan Anda, dan, jika ada, secara eksternal kepada pelanggan dan partner bisnis Anda.
- Tentukan proses permintaan akses darurat di sistem alur kerja permintaan layanan yang ada jika Anda memilikinya. Biasanya, sistem alur kerja semacam ini memungkinkan Anda membuat formulir penerimaan informasi untuk mengumpulkan informasi tentang permintaan, melacak permintaan melalui setiap tahap alur kerja, dan menambahkan langkah persetujuan otomatis dan manual. Hubungkan setiap permintaan dengan peristiwa darurat terkait yang dilacak dalam sistem manajemen insiden Anda. Dengan memiliki sistem yang seragam untuk akses darurat, Anda dapat melacak permintaan tersebut dalam sistem tunggal, menganalisis tren penggunaan, dan meningkatkan kualitas proses Anda.
- Pastikan proses akses darurat Anda hanya dapat dimulai oleh pengguna yang berwenang dan memerlukan persetujuan dari rekan sejawat atau manajemen pengguna yang sesuai. Proses persetujuan harus beroperasi secara efektif baik di dalam maupun di luar jam kerja. Tentukan bagaimana permintaan persetujuan mengizinkan pemberi persetujuan sekunder jika pemberi persetujuan utama tidak tersedia dan ditingkatkan ke rantai manajemen Anda hingga disetujui.
- Pastikan bahwa proses tersebut menghasilkan log dan peristiwa audit yang mendetail, baik untuk upaya yang berhasil maupun yang gagal untuk mendapatkan akses darurat. Pantau proses permintaan serta mekanisme akses darurat untuk mendeteksi penyalahgunaan atau akses yang tidak sah. Korelasikan aktivitas dengan peristiwa darurat yang sedang berlangsung dari sistem

manajemen insiden Anda dan munculkan peringatan ketika tindakan terjadi di luar periode waktu yang diharapkan. Misalnya, Anda harus memantau dan memperingatkan aktivitas dalam Akun AWS akses darurat, karena akun tersebut tidak boleh digunakan dalam operasi normal.

- Uji proses akses darurat secara berkala untuk memverifikasi bahwa langkah-langkahnya jelas dan memberikan tingkat akses yang benar dengan cepat dan efisien. Proses akses darurat Anda harus diuji sebagai bagian dari simulasi respons insiden (SEC10-BP07) dan tes pemulihan bencana (REL13-BP03).

Mode Kegagalan 1: Penyedia identitas yang digunakan untuk federasi ke AWS tidak tersedia

Seperti yang dijelaskan dalam SEC02-BP04 Mengandalkan penyedia identitas terpusat, kami sarankan Anda mengandalkan penyedia identitas terpusat untuk memfederasi pengguna tenaga kerja Anda untuk memberikan akses ke Akun AWS. Anda dapat melakukan federasi ke beberapa Akun AWS di organisasi AWS Anda menggunakan IAM Identity Center, atau Anda dapat melakukan federasi ke Akun AWS secara terpisah menggunakan IAM. Dalam kedua kasus tersebut, pengguna tenaga kerja melakukan autentikasi dengan penyedia identitas terpusat Anda sebelum diarahkan ke titik akhir masuk AWS ke masuk tunggal.

Apabila penyedia identitas terpusat Anda tidak tersedia, pengguna tenaga kerja Anda tidak dapat melakukan federasi ke Akun AWS atau mengelola beban kerja mereka. Dalam peristiwa darurat ini, Anda dapat menyediakan proses akses darurat untuk sekelompok kecil administrator untuk mengakses Akun AWS untuk melakukan tugas-tugas penting yang tidak dapat ditunda sampai penyedia identitas terpusat Anda kembali aktif. Misalnya, penyedia identitas Anda tidak tersedia selama 4 jam dan selama periode tersebut Anda perlu mengubah batas atas grup Amazon EC2 Auto Scaling di sebuah akun Produksi untuk menangani lonjakan lalu lintas pelanggan yang tidak terduga. Administrator darurat Anda harus mengikuti proses akses darurat untuk mendapatkan akses ke Akun AWS khusus produksi dan membuat perubahan yang diperlukan.

Proses akses darurat tersebut bergantung pada Akun AWS akses darurat yang telah dibuat sebelumnya yang digunakan semata-mata untuk akses darurat dan memiliki sumber daya AWS (seperti peran IAM danIAM users) untuk mendukung proses akses darurat. Selama operasi normal, tidak ada yang boleh mengakses akun akses darurat tersebut dan Anda harus memantau dan memperingatkan penyalahgunaan akun ini (untuk lebih jelasnya, lihat bagian panduan umum sebelumnya).

Akun akses darurat memiliki peran IAM akses darurat dengan izin untuk mengambil peran lintas akun di dalam Akun AWS yang memerlukan akses darurat. Peran IAM ini telah dibuat sebelumnya dan dikonfigurasi dengan kebijakan kepercayaan yang mempercayai peran IAM akun darurat.

Proses akses darurat dapat menggunakan salah satu pendekatan berikut:

- Anda dapat membuat satu set IAM users di awal untuk administrator darurat Anda di dalam akun akses darurat dengan kata sandi yang kuat dan token MFA terkait. Set IAM users ini memiliki izin untuk mengambil peran IAM yang kemudian memungkinkan akses lintas akun ke Akun AWS tempat akses darurat diperlukan. Kami sarankan Anda membuat pengguna sesedikit mungkin dan menetapkan setiap pengguna ke satu administrator darurat. Selama keadaan darurat, pengguna administrator darurat masuk ke akun akses darurat menggunakan kata sandi dan kode token MFA mereka, beralih ke peran IAM akses darurat di dalam akun darurat, dan akhirnya beralih ke peran IAM akses darurat di akun beban kerja untuk melakukan tindakan perubahan darurat. Kelebihan pendekatan ini adalah setiap IAM user ditugaskan ke satu administrator darurat dan Anda dapat mengetahui pengguna mana yang masuk dengan meninjau peristiwa CloudTrail. Kelemahannya adalah Anda harus mempertahankan beberapa IAM users dengan kata sandi berumur panjang dan token MFA yang terkait.
- Anda dapat menggunakan pengguna root Akun AWS akses darurat untuk masuk ke akun akses darurat, mengambil peran IAM untuk akses darurat, dan mengambil peran lintas akun di akun beban kerja. Kami merekomendasikan pengaturan kata sandi yang kuat dan beberapa token MFA untuk pengguna root. Kami juga menyarankan Anda menyimpan kata sandi dan token MFA di brankas kredensial korporasi aman yang memberlakukan autentikasi dan otorisasi yang kuat. Anda harus mengamankan kata sandi dan faktor pengaturan ulang token MFA: atur alamat email akun ke daftar distribusi email yang dipantau oleh administrator keamanan cloud Anda, dan nomor telepon akun ke nomor telepon bersama yang juga dipantau oleh administrator keamanan. Keunggulan pendekatan ini adalah ada satu set kredensial pengguna root untuk dikelola. Kelemahannya adalah karena ini merupakan pengguna bersama, beberapa administrator memiliki kemampuan untuk masuk sebagai pengguna root. Anda harus mengaudit peristiwa log brankas korporasi Anda untuk mengidentifikasi administrator mana yang menggunakan kata sandi pengguna root.

Mode Kegagalan 2: Konfigurasi penyedia identitas di AWS dimodifikasi atau telah kedaluwarsa

Agar pengguna tenaga kerja Anda dapat melakukan federasi ke Akun AWS, Anda dapat mengonfigurasi IAM Identity Center dengan penyedia identitas eksternal atau membuat Penyedia Identitas IAM (SEC02-BP04). Biasanya, Anda mengonfigurasinya dengan mengimpor dokumen XML metadata SAML yang disediakan oleh penyedia identitas Anda. Dokumen metadata XML tersebut mencakup sertifikat X.509 yang sesuai dengan kunci privat yang digunakan oleh penyedia identitas untuk menandatangani pernyataan SAML-nya.

Konfigurasi di sisi AWS ini dapat diubah atau dihapus secara tidak sengaja oleh administrator. Dalam skenario lain, sertifikat X.509 yang diimpor ke dalam AWS dapat kedaluwarsa dan XML metadata baru dengan sertifikat baru belum diimpor ke AWS. Kedua skenario ini dapat mengganggu federasi ke AWS untuk pengguna tenaga kerja Anda, yang mengakibatkan keadaan darurat.

Dalam keadaan darurat seperti ini, Anda dapat memberikan akses ke AWS kepada administrator identitas Anda untuk memperbaiki masalah federasi tersebut. Misalnya, administrator identitas Anda menggunakan proses akses darurat untuk masuk ke Akun AWS akses darurat, beralih ke peran di akun administrator Pusat Identitas, dan memperbarui konfigurasi penyedia identitas eksternal dengan mengimpor dokumen XML metadata SAML terbaru dari penyedia identitas Anda untuk mengaktifkan kembali federasi. Setelah federasi diperbaiki, pengguna tenaga kerja Anda melanjutkan penggunaan proses operasi normal untuk melakukan federasi ke akun beban kerja mereka.

Anda dapat mengikuti pendekatan yang dijelaskan dalam Mode Kegagalan 1 sebelumnya untuk membuat proses akses darurat. Anda dapat memberikan hak akses paling sedikit kepada administrator identitas Anda untuk mengakses hanya akun administrator Pusat Identitas dan melakukan tindakan pada Pusat Identitas di akun tersebut.

Mode Kegagalan 3: Gangguan Pusat Identitas

Apabila terjadi gangguan IAM Identity Center atau Wilayah AWS, kami sarankan Anda menyiapkan konfigurasi yang dapat Anda gunakan untuk menyediakan akses sementara ke AWS Management Console.

Proses akses darurat tersebut menggunakan federasi langsung dari penyedia identitas Anda ke IAM dalam akun darurat. Untuk detail tentang proses dan pertimbangan desain, lihat Menyiapkan akses darurat ke AWS Management Console.

Langkah implementasi

Langkah-langkah umum untuk semua mode kegagalan

- Buat Akun AWS yang ditujukan khusus untuk proses akses darurat. Di awal, buat sumber daya IAM yang dibutuhkan di dalam akun seperti peran IAM atau IAM users, dan Penyedia Identitas IAM opsional. Selain itu, buat di awal peran IAM lintas akun di dalam Akun AWS beban kerja dengan hubungan kepercayaan dengan IAM peran yang sesuai di akun akses darurat. Anda dapat menggunakan AWS CloudFormation StackSets dengan AWS Organizations untuk membuat sumber daya tersebut di akun anggota di dalam organisasi Anda.

- Buat AWS Organizations kebijakan kontrol layanan (SCP) untuk menyangkal penghapusan dan modifikasi peran IAM lintas akun di Akun AWS anggota.
- Aktifkan CloudTrail untuk Akun AWS akses darurat dan kirimkan peristiwa jejak ke bucket S3 pusat di Akun AWS pengumpulan log Anda. Jika Anda menggunakan AWS Control Tower untuk menyiapkan dan mengatur lingkungan multiakun AWS Anda, maka setiap akun yang Anda buat menggunakan AWS Control Tower atau daftarkan di AWS Control Tower memiliki CloudTrail yang diaktifkan secara default dan dikirim ke bucket S3 dalam Akun AWS arsip log khusus.
- Pantau aktivitas di akun akses darurat dengan membuat aturan EventBridge yang cocok saat login konsol dan aktivitas API berdasarkan peran IAM darurat. Kirimkan notifikasi ke pusat operasi keamanan Anda ketika aktivitas terjadi di luar peristiwa darurat yang sedang berlangsung yang dilacak dalam sistem manajemen insiden Anda.

Langkah-langkah tambahan untuk Mode Kegagalan 1: Penyedia identitas yang digunakan untuk melakukan federasi ke AWS tidak tersedia dan Mode Kegagalan 2: Konfigurasi penyedia identitas di AWS dimodifikasi atau telah kedaluwarsa

- Buat sumber daya di awal tergantung mekanisme yang Anda pilih untuk akses darurat:
  - Menggunakan IAM users buat IAM users di awal dengan kata sandi yang kuat serta perangkat MFA terkait.
  - Menggunakan pengguna root akun darurat: konfigurasikan pengguna root dengan kata sandi yang kuat dan simpan kata sandi di dalam brankas kredensial korporasi Anda. Kaitkan beberapa perangkat MFA fisik dengan pengguna root dan simpan perangkat di lokasi yang dapat diakses dengan cepat oleh anggota tim administrator darurat Anda.

Langkah-langkah tambahan untuk Mode Kegagalan 3: Gangguan pusat identitas

- Seperti yang dijelaskan dalam Menyiapkan akses darurat ke AWS Management Console, di Akun AWS akses darurat, buat sebuah Penyedia Identitas IAM untuk mengaktifkan federasi SAFL langsung dari penyedia identitas Anda.
- Buat grup operasi darurat di IdP Anda tanpa anggota.
- Buat peran IAM yang sesuai dengan grup operasi darurat di akun akses darurat.

Sumber daya

Praktik terbaik Well-Architected terkait:

- SEC02-BP04 Mengandalkan penyedia identitas terpusat
- SEC03-BP02 Memberikan hak akses paling rendah
- SEC10-BP02 Membuat rencana manajemen insiden
- SEC10-BP07 Menjalankan game day

Dokumen terkait:

- Menyiapkan akses darurat ke AWS Management Console
- Mengaktifkan pengguna federasi SAFL 2.0 untuk mengakses AWS Management Console
- Akses “pecah kaca”

Video terkait:

- AWS re:invent 2022 - Menyederhanakan akses tenaga kerja Anda dengan IAM Identity Center
- AWS re:Inforce 2022 - Pembahasan mendalam AWS Identity and Access Management (IAM)

Contoh terkait:

- Peran “Pecah Kaca” AWS
- Kerangka kerja playbook pelanggan AWS
- Contoh playbook respons insiden AWS

SEC03-BP04 Mengurangi izin secara terus-menerus

Jika tim Anda telah menentukan akses yang diperlukan, hapus izin yang tidak diperlukan dan tetapkan proses peninjauan untuk mendapatkan izin hak akses paling rendah. Pantau secara terus-menerus dan hapus identitas serta izin yang tidak diperlukan, baik untuk akses manusia maupun mesin.

Hasil yang diinginkan: Kebijakan izin harus mematuhi hak akses paling rendah. Setelah penetapan tugas dan peran pekerjaan sudah lebih baik, kebijakan izin Anda perlu ditinjau untuk menghapus izin yang tidak perlu. Pendekatan ini mempersempit cakupan dampak akibat kebocoran kredensial secara tidak sengaja, atau diakses tanpa otorisasi.

Antipola umum:

- Memberikan izin administrator kepada para pengguna secara default.
- Membuat kebijakan yang terlalu permisif, tetapi tanpa hak istimewa administrator penuh.
- Menyimpan kebijakan izin meski sudah tidak diperlukan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Setelah tim dan proyek mulai, kebijakan izin permisif mungkin digunakan untuk menumbuhkan inovasi dan ketangkasan. Misalnya, di dalam lingkungan pengembangan atau pengujian, developer dapat diberi akses ke seperangkat layanan AWS. Sebaiknya evaluasi akses secara terus-menerus dan batasi akses hanya untuk layanan dan tindakan layanan yang diperlukan untuk menyelesaikan tugas saat ini. Sebaiknya evaluasi ini dilakukan untuk identitas manusia maupun mesin. Identitas mesin, sering disebut sebagai akun layanan atau sistem, adalah identitas yang memberikan AWS akses ke aplikasi atau server. Akses ini penting terutama dalam lingkungan produksi, yang apabila izinnya terlalu permisif, dampaknya bisa luas dan berpotensi mengekspos data konsumen.

AWS menyediakan berbagai metode untuk membantu mengidentifikasi pengguna, peran, izin, dan kredensial yang tidak diperlukan. AWS juga dapat membantu menganalisis aktivitas akses oleh pengguna dan peran IAM, termasuk kunci akses terkait, dan akses ke sumber daya AWS, misalnya objek di bucket Amazon S3. Pembuatan kebijakan AWS Identity and Access Management Access Analyzer dapat membantu Anda menciptakan kebijakan pembatasan izin berdasarkan layanan dan tindakan aktual yang berinteraksi dengan pengguna utama. Kontrol akses berbasis atribut (ABAC) dapat membantu menyederhanakan manajemen izin. Dengan kontrol ini, Anda dapat memberikan izin kepada pengguna menggunakan atribut mereka tanpa perlu melampirkan kebijakan izin secara langsung ke setiap pengguna.

Langkah implementasi

- Gunakan AWS Identity and Access Management Access Analyzer: IAM Access Analyzer membantu mengidentifikasi sumber daya di organisasi dan akun Anda, seperti bucket Amazon Simple Storage Service (Amazon S3) atau peran IAM yang dibagikan kepada entitas eksternal.
- Gunakan pembuatan kebijakan IAM Access Analyzer: Pembuatan kebijakan IAM Access Analyzer membantu Anda membuat kebijakan izin terperinci berdasarkan aktivitas pengguna atau peran IAM.
- Tentukan rentang waktu yang diterima serta kebijakan penggunaan untuk pengguna dan peran IAM: Gunakan stempel waktu yang terakhir diakses untuk mengidentifikasi pengguna dan peran

yang tidak perlu lalu hapus. Tinjau informasi layanan dan tindakan yang terakhir diakses untuk mengidentifikasi dan menentukan cakupan izin bagi pengguna dan peran tertentu. Misalnya, Anda dapat menggunakan informasi yang terakhir diakses untuk mengidentifikasi tindakan Amazon S3 tertentu yang diperlukan oleh peran aplikasi dan membatasi akses hanya untuk tindakan tersebut. Fitur informasi yang terakhir diakses tersedia di AWS Management Console dan secara terprogram memungkinkan Anda menggabungkannya ke dalam alur kerja infrastruktur dan alat otomatis Anda.

- Pertimbangkan pencatatan log peristiwa data di AWS CloudTrail: Secara default, CloudTrail tidak mencatat log peristiwa data seperti aktivitas tingkat objek Amazon S3 (misalnya, `GetObject` dan `DeleteObject`) atau aktivitas tabel Amazon DynamoDB (misalnya, `PutItem` dan `DeleteItem`). Pertimbangkan untuk mengaktifkan pencatatan log pada peristiwa ini untuk menentukan pengguna dan peran apa yang perlu mengakses objek Amazon S3 dan item tabel DynamoDB tertentu.

Sumber daya

Dokumen terkait:

- Memberikan hak akses paling rendah
- Menghapus kredensial yang tidak diperlukan
- Apa itu AWS CloudTrail?
- Mengelola Kebijakan
- Pencatatan log dan pemantauan DynamoDB
- Mengaktifkan pencatatan log peristiwa CloudTrail untuk bucket dan objek Amazon S3
- Mendapatkan laporan kredensial untuk Akun AWS Anda

Video terkait:

- Menjadi Master Kebijakan IAM dalam 60 Menit atau Kurang
- Pemisahan Tugas, Hak Akses Paling Rendah, Delegasi, dan CI/CD
- AWS re:Inforce 2022 - Lebih dalam tentang AWS Identity and Access Management (IAM)

SEC03-BP05 Menentukan pagar pembatas izin untuk organisasi Anda

Tetapkan kontrol umum yang membatasi akses ke semua identitas di organisasi Anda. Misalnya, Anda dapat membatasi akses untuk Wilayah AWS tertentu, atau mencegah operator Anda

menghapus dari sumber daya umum, seperti IAM role yang digunakan untuk tim keamanan pusat Anda.

Antipola umum:

- Menjalankan beban kerja di akun administrator Organisasi Anda.
- Menjalankan beban kerja produksi dan non-produksi di akun yang sama.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Seiring Anda menumbuhkan dan mengelola beban kerja tambahan di AWS, Anda harus memisahkan semua beban kerja ini menggunakan akun dan mengelola akun-akun tersebut menggunakan AWS Organizations. Kami menyarankan Anda membuat pagar pembatas izin umum yang membatasi akses ke semua identitas di organisasi Anda. Misalnya, Anda dapat membatasi akses ke Wilayah AWS tertentu, atau mencegah tim Anda menghapus sumber daya umum, seperti IAM role yang digunakan oleh tim keamanan pusat Anda.

Anda dapat memulainya dengan mengimplementasikan contoh kebijakan kontrol layanan, seperti mencegah pengguna menonaktifkan layanan utama. SCP menggunakan bahasa kebijakan IAM dan memungkinkan Anda untuk menerapkan kontrol yang dipatuhi semua principal (pengguna dan peran). Anda dapat membatasi akses ke tindakan atau sumber daya layanan tertentu, dan berdasarkan kondisi tertentu untuk memenuhi kebutuhan kontrol akses organisasi Anda. Jika perlu, Anda dapat menetapkan pengecualian pada pagar pembatas Anda. Misalnya, Anda dapat membatasi tindakan layanan untuk semua entitas IAM di dalam akun kecuali untuk peran administrator tertentu.

Kami menyarankan Anda untuk tidak menjalankan beban kerja di akun manajemen Anda. Akun manajemen sebaiknya digunakan untuk menata kelola dan men-deploy pagar pembatas keamanan yang akan memengaruhi akun-akun anggota. Beberapa layanan AWS mendukung penggunaan akun administrator yang didelegasikan. Saat tersedia, Anda harus menggunakan akun delegasi ini sebagai pengganti akun manajemen. Anda harus membatasi secara ketat akses ke akun administrator Organisasi.

Menggunakan strategi multi-akun memungkinkan Anda untuk memiliki fleksibilitas yang lebih besar dalam menerapkan pagar pembatas ke beban kerja Anda. Arsitektur Rujukan Keamanan AWS memberikan panduan preskriptif tentang cara merancang struktur akun Anda. Layanan AWS seperti AWS Control Tower menyediakan kemampuan untuk mengelola kontrol preventif serta detektif

secara terpusat di seluruh organisasi Anda. Tetapkan tujuan yang jelas untuk tiap akun atau OU di dalam organisasi dan batasi kontrol yang sejalan dengan tujuan tersebut.

Sumber daya

Dokumen terkait:

- AWS Organizations
- Kebijakan kontrol layanan (SCP)
- Dapatkan hasil maksimal dari kebijakan kontrol layanan di lingkungan multi-akun
- Arsitektur Referensi Keamanan AWS (AWS SRA)

Video terkait:

- Tegakkan Pagar Pembatas Preventif menggunakan Kebijakan Kontrol Layanan
- Membangun tata kelola pada skala besar dengan AWS Control Tower
- Mendalami AWS Identity and Access Management

SEC03-BP06 Mengelola akses berdasarkan siklus hidup

Integrasikan kontrol akses dengan siklus hidup operator dan aplikasi serta penyedia federasi terpusat. Misalnya, hapus akses pengguna saat mereka keluar dari organisasi atau berganti peran.

Saat Anda mengelola beban kerja menggunakan akun terpisah, akan ada kasus saat Anda perlu membagikan sumber daya kepada akun-akun tersebut. Sebaiknya bagikan sumber menggunakan AWS Resource Access Manager (AWS RAM). Layanan ini memungkinkan Anda untuk membagikan sumber daya AWS di dalam Unit Organisasi dan AWS Organizations Anda. Menggunakan AWS RAM, akses ke sumber daya bersama secara otomatis diberikan atau dicabut ketika akun dimasukkan atau dikeluarkan dari Organisasi atau Unit Organisasi mereka. Hal ini memastikan bahwa sumber daya hanya dibagikan dengan akun yang Anda maksudkan saja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Siklus hidup akses pengguna mengimplementasikan kebijakan siklus hidup akses pengguna untuk pengguna yang baru bergabung, perubahan fungsi tugas, serta pengguna yang keluar, sehingga hanya pengguna saat ini yang memiliki akses.

Sumber daya

Dokumen terkait:

- Kontrol akses berbasis atribut (ABAC)
- Berikan hak akses paling rendah
- IAM Access Analyzer
- Hapus kredensial yang tidak diperlukan
- Bekerja dengan Kebijakan

Video terkait:

- Become an IAM Policy Master in 60 Minutes or Less
- Separation of Duties, Least Privilege, Delegation, and CI/CD

SEC03-BP07 Menganalisis akses lintas akun dan publik

Pantau secara terus-menerus temuan yang menyoroti akses lintas akun dan publik. Kurangi akses publik dan akses lintas akun hanya ke sumber daya yang memerlukan akses ini.

Hasil yang diinginkan: Mengetahui mana sumber daya AWS yang dapat dibagikan dan kepada siapa. Pantau secara terus-menerus dan audit sumber daya bersama untuk memastikan sumber daya tersebut hanya dibagikan kepada pengguna utama yang sah.

Antipola umum:

- Tidak menyimpan inventaris sumber daya bersama.
- Tidak mengikuti proses persetujuan akses lintas akun atau publik ke sumber daya.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Rendah

Panduan implementasi

Jika akun Anda berada di AWS Organizations, Anda dapat memberikan akses sumber daya ke seluruh organisasi, unit organisasi tertentu, atau akun individu. Jika akun Anda bukan anggota suatu organisasi, Anda dapat berbagi sumber daya dengan akun individu. Anda dapat memberikan akses lintas akun langsung menggunakan kebijakan berbasis sumber daya — contohnya, kebijakan bucket

Amazon Simple Storage Service (Amazon S3) — atau dengan mengizinkan pengguna utama di akun lain menggunakan suatu peran IAM di akun Anda. Saat menggunakan kebijakan sumber daya, pastikan bahwa akses tersebut hanya diberikan kepada pengguna utama yang sah. Tentukan proses untuk menyetujui semua sumber daya yang diperlukan untuk tersedia secara publik.

AWS Identity and Access Management Access Analyzer menggunakan keamanan yang dapat dibuktikan untuk mengidentifikasi semua jalur akses ke sumber daya dari luar akunnya. Keamanan tersebut meninjau kebijakan sumber daya secara terus-menerus, dan melaporkan temuan akses lintas akun dan publik untuk memudahkan Anda menganalisis potensi akses yang meluas. Pertimbangkan untuk mengonfigurasi IAM Access Analyzer dengan AWS Organizations untuk memastikan Anda memiliki visibilitas ke semua akun Anda. IAM Access Analyzer juga mendukung Anda untuk melakukan pratinjau temuan sebelum melakukan deployment izin sumber daya. Hal ini memungkinkan Anda untuk memvalidasi bahwa perubahan kebijakan hanya memberikan akses lintas akun dan publik tertentu ke sumber daya Anda. Saat merancang akses multiakun, Anda dapat menggunakan kebijakan kepercayaan untuk mengontrol dalam kasus seperti apa suatu peran bisa didapatkan. Misalnya, Anda dapat menggunakan kunci kondisi `PrincipalOrgId` untuk menolak upaya untuk mendapatkan peran dari luar AWS Organizations Anda.

AWS Config dapat melaporkan sumber daya yang konfigurasinya salah, dan melalui pemeriksaan kebijakan AWS Config, dapat mendeteksi sumber daya dengan konfigurasi akses publik. Layanan seperti AWS Control Tower dan AWS Security Hub menyederhanakan deployment kontrol deteksi dan pagar pembatas di seluruh AWS Organizations untuk mengidentifikasi dan memulihkan sumber daya yang terekspos ke publik. Misalnya, AWS Control Tower memiliki pagar pembatas terkelola yang dapat mendeteksi adanya snapshot Amazon EBS yang dapat dipulihkan di Akun AWS.

Langkah implementasi

- Pertimbangkan untuk mengaktifkan AWS Config untuk AWS Organizations: AWS Config memungkinkan Anda mengumpulkan temuan dari banyak akun di dalam satu AWS Organizations ke akun administrator yang ditunjuk. Layanan ini memberikan tampilan komprehensif dan membantu Anda melakukan deployment Aturan AWS Config di seluruh akun untuk mendeteksi sumber daya yang dapat diakses publik.
- Konfigurasikan AWS Identity and Access Management Access Analyzer: IAM Access Analyzer membantu Anda mengidentifikasi sumber daya di organisasi dan akun Anda, seperti bucket Amazon S3 atau peran IAM yang dibagikan kepada entitas eksternal.
- Gunakan perbaikan otomatis di AWS Config untuk merespons perubahan dalam konfigurasi akses publik di bucket Amazon S3: Anda dapat secara otomatis mengaktifkan kembali pengaturan blokir akses publik untuk bucket Amazon S3.

- Implementasikan pemantauan dan peringatan untuk mengidentifikasi apakah Amazon S3 dapat diakses publik: Anda harus menerapkan pemantauan dan peringatan untuk mengidentifikasi saat Amazon S3 Blokir Akses Publik dinonaktifkan, dan saat bucket Amazon S3 dapat diakses publik. Selain itu, jika Anda menggunakan AWS Organizations, Anda dapat membuat kebijakan kontrol layanan yang mencegah perubahan pada kebijakan akses publik Amazon S3. AWS Trusted Advisor memeriksa apakah ada bucket Amazon S3 yang memiliki izin akses terbuka. Izin bucket yang memberikan, mengunggah, atau menghapus akses ke semua orang akan menciptakan potensi masalah keamanan dengan mengizinkan siapa pun untuk menambahkan, mengubah, atau menghapus item dalam bucket. Pemeriksaan Trusted Advisor memeriksa izin bucket eksplisit dan kebijakan bucket terkait yang mungkin mengganti izin bucket. Anda juga dapat menggunakan AWS Config untuk memantau bucket Amazon S3 Anda untuk akses publik. Untuk informasi selengkapnya, kunjungi Cara Menggunakan AWS Config untuk Memantau dan Merespons Bucket Amazon S3 yang Mengizinkan Akses Publik. Saat meninjau akses, penting untuk mengetahui jenis data yang ada di bucket Amazon S3. Amazon Macie membantu menemukan dan melindungi data sensitif seperti PII, PHI, dan kredensial seperti kunci AWS atau privat.

Sumber daya

Dokumen terkait:

- Menggunakan AWS Identity and Access Management Access Analyzer
- Pustaka kontrol AWS Control Tower
- Standar Praktik Terbaik Keamanan Dasar AWS
- Aturan Terkelola AWS Config
- Referensi pemeriksaan AWS Trusted Advisor
- Memantau hasil pemeriksaan AWS Trusted Advisor dengan Amazon EventBridge
- Mengelola Aturan AWS Config Seluruh Akun di Organisasi Anda
- AWS Config dan AWS Organizations

Video terkait:

- Praktik Terbaik untuk mengamankan lingkungan multiakun Anda
- Memahami IAM Access Analyzer Lebih Dalam

SEC03-BP08 Membagikan sumber daya secara aman dalam organisasi Anda

Seiring meningkatnya jumlah beban kerja, Anda mungkin perlu membagikan akses ke sumber daya dalam beban kerja tersebut atau berulang kali menyediakan sumber daya tersebut di seluruh akun. Anda mungkin memiliki konsep untuk membagi lingkungan Anda dalam beberapa kelompok, seperti lingkungan pengembangan, pengujian, dan produksi. Konsep pemisahan ini tidak akan membatasi Anda untuk berbagi secara aman. Dengan membagikan komponen yang tumpang tindih, Anda dapat mengurangi overhead operasional dan memungkinkan pengalaman yang konsisten tanpa ada yang terlewatkan sambil membuat sumber daya yang sama berulang kali.

Hasil yang diinginkan: Meminimalkan akses yang tidak diinginkan menggunakan metode yang aman untuk berbagi sumber daya dalam organisasi, dan membantu inisiatif pencegahan kehilangan data. Mengurangi overhead operasional daripada mengelola komponen satu per satu, yang akan mengurangi kesalahan dari pembuatan komponen yang sama secara manual berulang kali, serta meningkatkan skalabilitas beban kerja. Anda dapat memperoleh manfaat dari pengurangan waktu hingga resolusi di skenario kegagalan multi-titik, dan meningkatkan keyakinan Anda dalam menentukan kapan komponen tidak diperlukan lagi. Untuk panduan preskriptif dalam menganalisis sumber daya yang dibagikan secara eksternal, lihat SEC03-BP07 Menganalisis akses lintas akun dan publik.

Antipola umum:

- Tidak ada proses untuk terus memantau dan memberikan peringatan otomatis terkait pembagian secara eksternal yang tidak terduga.
- Tidak ada acuan terkait apa yang boleh dan tidak boleh dibagikan.
- Kebijakan terbuka luas secara default, bukannya berbagi secara eksplisit ketika diperlukan.
- Membuat sumber daya dasar secara manual, yang tumpang tindih saat diperlukan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Rancang pola dan kontrol akses Anda untuk mengelola penggunaan sumber daya yang dibagikan secara aman dan hanya dengan entitas tepercaya. Pantau sumber daya yang dibagikan dan tinjau akses sumber daya yang dibagikan secara terus-menerus serta tetap waspada terhadap pembagian yang tidak terduga atau tidak tepat. Lihat Menganalisis akses lintas akun dan akses publik untuk

membantu Anda menetapkan tata kelola guna mengurangi akses eksternal hanya ke sumber daya yang memerlukannya, dan menetapkan proses untuk terus memantau dan memberikan peringatan secara otomatis.

Berbagi lintas akun dalam AWS Organizations didukung oleh sejumlah layanan AWS, seperti AWS Security Hub, Amazon GuardDuty, dan AWS Backup. Layanan tersebut memungkinkan data untuk dibagikan ke akun pusat, dapat diakses dari akun pusat, atau mengelola sumber daya dan data dari akun pusat. Misalnya, AWS Security Hub dapat mentransfer temuan dari akun individu ke akun pusat sehingga Anda dapat melihat semua temuan. AWS Backup dapat melakukan pencadangan untuk sumber daya dan membagikannya ke seluruh akun. Anda dapat menggunakan AWS Resource Access Manager (AWS RAM) untuk membagikan sumber daya umum, seperti subnet VPC dan lampiran Transit Gateway, AWS Network Firewall, atau jalur Amazon SageMaker.

Untuk membatasi akun Anda agar hanya berbagi sumber daya dalam organisasi, gunakan kebijakan kontrol layanan (SCP) untuk mencegah akses ke pengguna utama eksternal. Saat membagikan sumber daya, kombinasikan kontrol berbasis identitas dan kontrol jaringan untuk membuat perimeter data bagi organisasi Anda guna membantu melindungi dari akses yang tidak diinginkan. Perimeter data adalah kumpulan pagar pembatas preventif untuk membantu memverifikasi bahwa hanya identitas yang Anda percaya yang mengakses sumber daya tepercaya dari jaringan yang dikenal. Kontrol ini menetapkan batas yang sesuai terkait sumber daya apa yang dapat dibagikan, serta mencegah dibagikan atau bocornya sumber daya yang tidak semestinya dibagikan. Misalnya, sebagai bagian dari perimeter data, Anda dapat menggunakan kebijakan titik akhir VPC dan persyaratan `:PrincipalOrgId AWS` untuk memastikan bahwa identitas yang mengakses bucket Amazon S3 adalah milik organisasi Anda. Penting diketahui bahwa SCP tidak berlaku untuk peran terkait layanan (LSR) atau pengguna utama layanan AWS.

Saat menggunakan Amazon S3, nonaktifkan ACL untuk bucket Amazon S3 Anda dan gunakan kebijakan IAM untuk menentukan kontrol akses. Untuk membatasi akses ke Amazon S3 asal dari Amazon CloudFront, migrasikan identitas akses asal (OAI) ke kontrol akses awal (OAC) yang mendukung fitur tambahan, termasuk enkripsi di sisi server dengan AWS Key Management Service.

Dalam beberapa kasus, Anda mungkin ingin mengizinkan pembagian sumber daya ke luar organisasi Anda atau memberikan pihak ketiga akses ke sumber daya Anda. Untuk panduan preskriptif tentang manajemen izin untuk membagikan sumber daya secara eksternal, lihat Manajemen izin.

Langkah implementasi

1. Gunakan AWS Organizations.

AWS Organizations adalah layanan manajemen akun yang memungkinkan Anda untuk menggabungkan beberapa Akun AWS ke dalam organisasi yang Anda buat dan kelola secara terpusat. Anda dapat mengelompokkan akun ke dalam unit organisasi (OU) dan melampirkan kebijakan yang berbeda ke setiap OU untuk membantu memenuhi kebutuhan anggaran, keamanan, dan kepatuhan. Anda juga dapat mengontrol cara layanan kecerdasan buatan (AI) dan machine learning (ML) AWS mengumpulkan dan menyimpan data, serta menggunakan manajemen multiakun layanan AWS yang terintegrasi dengan Organizations.

2. Integrasikan AWS Organizations dengan layanan AWS.

   Saat Anda mengaktifkan layanan AWS untuk melakukan tugas atas nama Anda di akun anggota organisasi, AWS Organizations membuat peran terkait layanan IAM untuk layanan tersebut di setiap akun anggota. Anda harus mengelola akses tepercaya menggunakan AWS Management Console, API AWS, atau AWS CLI. Untuk panduan preskriptif tentang mengaktifkan akses tepercaya, lihat Menggunakan AWS Organizations dengan layanan AWS lainnya dan Layanan AWS yang dapat digunakan dengan Organizations.

3. Buat perimeter data.

   Perimeter AWS biasanya direpresentasikan sebagai organisasi yang dikelola oleh AWS Organizations. Selain sistem dan jaringan on-premise, mengakses sumber daya AWS adalah hal yang banyak orang kategorikan sebagai salah satu perimeter AWS Saya. Perimeter bertujuan untuk memverifikasi bahwa akses diizinkan jika identitas dipercaya, sumber daya dipercaya, dan jaringan dikenal.

   a. Menentukan dan mengimplementasikan perimeter.

      Ikuti langkah yang dijelaskan dalam Implementasi perimeter dalam Membangun Perimeter di laporan resmi AWS untuk setiap persyaratan otorisasi. Untuk panduan preskriptif tentang melindungi lapisan jaringan, lihat Melindungi jaringan.

   b. Tetap pantau dan waspada.

      AWS Identity and Access Management Access Analyzer membantu mengidentifikasi sumber daya di organisasi Anda dan akun yang dibagikan kepada entitas eksternal. Anda dapat mengintegrasikan IAM Access Analyzer dengan AWS Security Hub guna mengirimkan dan menggabungkan temuan untuk sumber daya dari IAM Access Analyzer ke Security Hub untuk membantu menganalisis postur keamanan lingkungan Anda. Untuk mengaktifkan integrasi, aktifkan IAM Access Analyzer dan Security Hub di setiap Wilayah di setiap akun. Anda juga dapat menggunakan Aturan AWS Config untuk mengaudit konfigurasi dan memberikan

peringatan kepada pihak yang sesuai menggunakan AWS Chatbot dengan AWS Security Hub. Anda kemudian dapat menggunakan dokumen AWS Systems Manager Automation untuk memperbaiki sumber daya yang melanggar kepatuhan.

c. Untuk panduan preskriptif tentang pemantauan dan peringatan secara berkelanjutan terkait sumber daya yang dibagikan secara eksternal, lihat Menganalisis akses lintas akun dan publik.

4. Gunakan fitur berbagi sumber daya di layanan AWS dan batasi sesuai kebutuhan.

   Banyak layanan AWS dapat Anda gunakan untuk membagikan sumber daya kepada akun lainnya, atau menargetkan sumber daya di akun lainnya, seperti Amazon Machine Images (AMI) dan AWS Resource Access Manager (AWS RAM). Batasi API `ModifyImageAttribute` guna menentukan akun tepercaya untuk berbagi AMI. Tentukan persyaratan `ram:RequestedAllowsExternalPrincipals` saat menggunakan AWS RAM untuk membatasi berbagi hanya ke organisasi Anda, untuk membantu mencegah akses dari identitas yang tidak tepercaya. Untuk panduan preskriptif dan pertimbangan, lihat Berbagi sumber daya dan target eksternal.

5. Gunakan AWS RAM untuk berbagi dengan aman dalam akun atau dengan Akun AWS lainnya.

   AWS RAM membantu Anda secara aman membagikan sumber daya yang Anda buat kepada peran dan pengguna di akun Anda serta kepada Akun AWS lainnya. Dalam lingkungan multiakun, AWS RAM memungkinkan Anda untuk membuat sumber daya satu kali dan membagikannya kepada akun lain. Pendekatan ini membantu mengurangi overhead operasional sekaligus memberikan konsistensi, visibilitas, dan auditabilitas melalui integrasi dengan Amazon CloudWatch dan AWS CloudTrail, yang tidak Anda dapatkan saat menggunakan akses lintas akun.

   Jika ada sumber daya yang sebelumnya dibagikan menggunakan kebijakan berbasis sumber daya, Anda dapat menggunakan API `PromoteResourceShareCreatedFromPolicy` atau yang setara untuk mendukung berbagi sumber daya ke berbagi sumber daya AWS RAM penuh.

   Dalam beberapa kasus, Anda mungkin memerlukan beberapa langkah tambahan untuk berbagi sumber daya. Misalnya, untuk membagikan snapshot terenkripsi, Anda perlu membagikan kunci AWS KMS.

Sumber daya

Praktik Terbaik Terkait:

- SEC03-BP07 Menganalisis akses lintas akun dan publik
- SEC03-BP09 Membagikan sumber daya secara aman kepada pihak ketiga
- SEC05-BP01 Membuat lapisan jaringan

Dokumen terkait:

- Pemilik bucket yang memberikan izin lintas akun ke objek yang tidak dimilikinya
- Cara menggunakan Kebijakan Kepercayaan dengan IAM
- Membangun Perimeter Data di AWS
- Cara menggunakan ID eksternal saat memberikan akses ke sumber daya AWSAnda kepada pihak ketiga
- Layanan AWS yang dapat digunakan dengan AWS Organizations
- Membuat perimeter data di AWS: Izinkan hanya identitas tepercaya untuk mengakses data perusahaan

Video terkait:

- Akses Terperinci dengan AWS Resource Access Manager
- Mengamankan perimeter data Anda dengan titik akhir VPC
- Membuat perimeter data di AWS

Alat terkait:

- Contoh Kebijakan Perimeter Data

SEC03-BP09 Membagikan sumber daya secara aman kepada pihak ketiga

Keamanan lingkungan cloud tidak berhenti di organisasi Anda. Organisasi Anda mungkin menggunakan pihak ketiga untuk mengelola sebagian data Anda. Manajemen izin untuk sistem yang dikelola pihak ketiga harus mengikuti praktik akses sesuai kebutuhan menggunakan prinsip hak akses paling rendah dengan kredensial sementara. Melalui kerja sama dengan pihak ketiga, Anda dapat mengurangi cakupan dampak sekaligus risiko dari akses yang tidak diinginkan.

Hasil yang diinginkan: Kredensial AWS Identity and Access Management (IAM) jangka panjang, kunci akses IAM, dan kunci rahasia yang terkait dengan pengguna dapat digunakan oleh siapa saja

selama kredensialnya valid dan aktif. Menggunakan peran IAM dan kredensial sementara membantu Anda meningkatkan kekukuhan keamanan dengan mengurangi upaya manajemen kredensial jangka panjang, termasuk manajemen dan overhead operasional terkait detail sensitif tersebut. Dengan pengidentifikasi unik universal (UUID) untuk ID eksternal dalam kebijakan kepercayaan IAM, dan menjaga kebijakan IAM untuk peran IAM di bawah kendali Anda, Anda dapat mengaudit dan memverifikasi bahwa akses yang diberikan kepada pihak ketiga tidak terlalu permisif. Untuk panduan preskriptif dalam menganalisis sumber daya yang dibagikan secara eksternal, lihat SEC03-BP07 Menganalisis akses lintas akun dan publik.

Antipola umum:

- Menggunakan kebijakan kepercayaan IAM default tanpa persyaratan apa pun.
- Menggunakan kunci akses dan kredensial IAM jangka panjang.
- Menggunakan kembali ID eksternal.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Anda mungkin ingin mengizinkan pembagian sumber daya ke luar AWS Organizations atau memberi pihak ketiga akses ke akun Anda. Misalnya, pihak ketiga mungkin menyediakan solusi pemantauan yang perlu mengakses sumber daya di akun Anda. Dalam kasus tersebut, buat peran lintas akun IAM yang hanya memiliki hak akses sesuai yang dibutuhkan oleh pihak ketiga tersebut. Selain itu, tentukan kebijakan kepercayaan menggunakan persyaratan ID eksternal. Saat menggunakan ID eksternal, Anda atau pihak ketiga dapat membuat ID unik untuk setiap pelanggan, pihak ketiga, atau penghunian. Setelah dibuat, ID unik tidak boleh dikontrol siapa pun selain Anda. Pihak ketiga harus mengimplementasikan proses untuk memberikan ID eksternal melalui cara yang aman, dapat diaudit, dan diproduksi kembali.

Anda juga dapat menggunakan IAM Roles Anywhere guna mengelola peran IAM untuk aplikasi di luar AWS yang menggunakan API AWS.

Hapus peran tersebut jika pihak ketiga sudah tidak perlu mengakses lingkungan Anda. Hindari menyediakan kredensial jangka panjang kepada pihak ketiga. Selalu waspadai layanan AWS lainnya yang mendukung fitur berbagi. Misalnya, AWS Well-Architected Tool memungkinkan berbagi beban kerja dengan Akun AWS lainnya, dan AWS Resource Access Manager membantu membagikan sumber daya AWS yang Anda miliki secara aman kepada akun lain.

Langkah implementasi

1. Gunakan peran lintas akun untuk memberikan akses kepada akun eksternal.

   Peran lintas akun mengurangi jumlah informasi sensitif yang disimpan oleh akun eksternal dan pihak ketiga untuk memberikan layanan kepada pelanggannya. Peran lintas akun memungkinkan Anda untuk memberikan akses ke sumber daya AWS di akun Anda kepada pihak ketiga secara aman, seperti AWS Partner atau akun lainnya di organisasi Anda, dan Anda pun tetap dapat mengelola dan mengaudit akses tersebut.

   Pihak ketiga mungkin memberikan layanan kepada Anda dari infrastruktur hibrida atau menarik data ke lokasi di luar situs. IAM Roles Anywhere membantu Anda memungkinkan beban kerja pihak ketiga berinteraksi dengan beban kerja AWS Anda secara aman dan makin mengurangi kebutuhan kredensial jangka panjang.

   Anda tidak boleh menggunakan kredensial jangka panjang, atau kunci akses yang terkait dengan pengguna, untuk menyediakan akses kepada akun eksternal. Sebaiknya gunakan peran lintas akun untuk memberikan akses lintas akun.

2. Gunakan ID eksternal dengan pihak ketiga.

   Menggunakan ID eksternal memungkinkan Anda untuk menunjuk siapa yang dapat mengambil peran di kebijakan kepercayaan IAM. Kebijakan kepercayaan mungkin mengharuskan pengguna yang mengambil peran menegaskan persyaratan dan target operasi. Dengan cara ini, pemilik akun dapat mengizinkan peran tersebut untuk diambil hanya dalam keadaan tertentu. Fungsi utama ID eksternal adalah untuk mencegah dan menangani masalah confused deputy.

   Gunakan ID eksternal jika Anda adalah pemilik Akun AWS dan sudah mengonfigurasi peran untuk pihak ketiga yang mengakses Akun AWS lainnya selain akun Anda, atau jika Anda mengambil peran atas nama pelanggan yang lain. Jalin kerja sama dengan pihak ketiga atau AWS Partner untuk menentukan persyaratan ID eksternal yang akan disertakan dalam kebijakan kepercayaan IAM.

3. Gunakan ID eksternal yang unik secara universal.

   Implementasikan proses yang membuat nilai unik acak yang unik untuk ID eksternal, seperti pengidentifikasi unik universal (UUID). Pihak ketiga yang menggunakan kembali ID eksternal untuk pengguna yang berbeda tidak menangani masalah confused deputy karena pelanggan A mungkin dapat melihat data pelanggan B menggunakan peran ARN pelanggan B serta duplikat ID eksternal. Dalam lingkungan multipenyewa yang di dalamnya ada pihak ketiga yang mendukung beberapa pelanggan dengan Akun AWS yang berbeda, pihak ketiga tersebut harus menggunakan ID unik yang berbeda sebagai ID eksternal untuk setiap Akun AWS. Pihak ketiga bertanggung

jawab untuk mendeteksi duplikat ID eksternal dan memetakan setiap pelanggan secara aman ke ID eksternal masing-masing. Pihak ketiga harus menguji untuk memverifikasi bahwa pihaknya hanya dapat mengambil peran saat menentukan ID eksternal. Pihak ketiga dilarang menyimpan ARN peran pelanggan dan ID eksternal hingga ID eksternal diperlukan.

ID eksternal bukan sesuatu yang rahasia, tetapi tidak boleh berupa nilai yang mudah ditebak, seperti nomor telepon, nama, atau ID akun. Buat ID eksternal menjadi bidang hanya baca sehingga ID eksternal tidak dapat diubah untuk tujuan meniru penyiapan.

Anda atau pihak ketiga dapat membuat ID eksternal. Bentuk proses untuk menentukan siapa yang bertanggung jawab dalam pembuatan ID. Siapa pun entitas pembuat ID eksternalnya, pihak ketiga menjaga keunikan dan formatnya tetap konsisten untuk semua pelanggan.

4. Hentikan kredensial jangka panjang yang disediakan pelanggan.

   Hentikan penggunaan kredensial jangka panjang dan gunakan peran lintas akun atau IAM Roles Anywhere. Jika Anda harus menggunakan kredensial jangka panjang, buat rencana atau migrasikan ke akses berbasis peran. Untuk detail tentang manajemen kunci, lihat Manajemen Identitas. Selain itu, jalin kerja sama dengan tim Akun AWS Anda dan pihak ketiga untuk menyusun runbook mitigasi risiko. Untuk panduan preskriptif tentang merespons dan memitigasi potensi dampak insiden keamanan, lihat Respons insiden.

5. Verifikasi bahwa penyiapan memiliki panduan preskriptif atau diotomatisasi.

   Kebijakan yang dibuat untuk akses lintas akun di akun Anda harus mematuhi prinsip hak akses paling rendah. Pihak ketiga harus menyediakan dokumen kebijakan peran atau mekanisme penyiapan otomatis yang menggunakan templat AWS CloudFormation atau yang setara. Hal ini mengurangi potensi kesalahan yang bisa terjadi pada pembuatan kebijakan manual dan menyediakan jejak yang dapat diaudit. Untuk informasi lebih lanjut tentang menggunakan templat AWS CloudFormation untuk membuat peran lintas akun, lihat Peran Lintas Akun.

   Pihak ketiga harus menyediakan mekanisme penyiapan otomatis yang dapat diaudit. Namun, dengan dokumen kebijakan peran yang menguraikan akses yang diperlukan, Anda harus mengotomatiskan penyiapan peran. Anda harus memantau perubahan dengan deteksi penyimpangan menggunakan templat AWS CloudFormation atau yang setara sebagai bagian dari praktik audit.

6. Antisipasi perubahan.

   Struktur akun Anda, kebutuhan Anda akan pihak ketiga, atau penawaran layanan yang disediakan dapat berubah. Anda harus mengantisipasi perubahan dan kegagalan, dan membuat rencana

yang sesuai dengan orang, proses, dan teknologi yang tepat. Audit tingkat akses yang Anda berikan secara berkala, dan terapkan metode deteksi untuk memberi tahu Anda tentang perubahan yang tidak terduga. Pantau dan audit penggunaan peran dan penyimpanan data ID eksternal. Anda harus bersiap untuk mencabut akses pihak ketiga, baik untuk sementara atau secara permanen, jika ada perubahan atau pola akses yang tidak terduga. Selain itu, ukur dampak atas operasi pencabutan Anda, termasuk waktu yang diperlukan untuk melakukannya, orang yang terlibat, biaya, dan dampak terhadap sumber daya lainnya.

Untuk panduan preskriptif tentang metode deteksi, lihat Praktik terbaik deteksi.

Sumber daya

Praktik Terbaik Terkait:

- SEC02-BP02 Menggunakan kredensial sementara
- SEC03-BP05 Menentukan pagar pembatas izin untuk organisasi Anda
- SEC03-BP06 Mengelola akses berdasarkan siklus hidup
- SEC03-BP07 Menganalisis akses lintas akun dan publik
- SEC04 Deteksi

Dokumen terkait:

- Pemilik bucket yang memberikan izin lintas akun ke objek yang tidak dimilikinya
- Cara menggunakan kebijakan kepercayaan dengan peran IAM
- Mendelegasikan akses di seluruh Akun AWS menggunakan peran IAM
- Bagaimana cara mengakses sumber daya di Akun AWS lainnya menggunakan IAM?
- Praktik terbaik keamanan dalam IAM
- Logika evaluasi kebijakan lintas akun
- Cara menggunakan ID eksternal saat memberikan akses ke sumber daya AWS Anda kepada pihak ketiga
- Mengumpulkan Informasi dari Sumber Daya AWS CloudFormation yang Dibuat di Akun Eksternal dengan Sumber Daya Kustom
- Menggunakan ID Eksternal Secara Aman untuk Mengakses Akun AWS Milik Pihak Lain
- Memperluas peran IAM ke beban kerja di luar IAM dengan IAM Roles Anywhere

Video terkait:

- Bagaimana caranya mengizinkan pengguna atau peran di Akun AWS yang terpisah untuk mengakses Akun AWS saya?
- AWS re:Invent 2018: Menjadi Master Kebijakan IAM dalam 60 Menit atau Kurang
- Pusat Pengetahuan AWS Live: Praktik Terbaik IAM dan Keputusan Rancangan

Contoh terkait:

- Well-Architected Lab - Pengambilan peran IAM lintas akun Lambda (Level 300)
- Mengonfigurasi akses lintas akun ke Amazon DynamoDB
- AWS STS Network Query Tool

Deteksi

Deteksi terdiri dari dua bagian: deteksi perubahan konfigurasi yang tidak diinginkan atau tidak diharapkan, dan deteksi perilaku yang tidak diharapkan. Deteksi yang pertama dapat dilakukan di beberapa tempat dalam siklus hidup pengiriman aplikasi. Menggunakan infrastruktur sebagai kode (misalnya, templat CloudFormation), Anda dapat memeriksa konfigurasi yang tidak diinginkan sebelum melakukan deployment beban kerja dengan mengimplementasikan pemeriksaan dalam pipeline CI/CD atau kontrol sumber. Lalu, seiring dengan deployment beban kerja ke lingkungan produksi dan nonproduksi, Anda dapat memeriksa konfigurasi menggunakan AWS asli, sumber terbuka, atau alat Partner AWS. Pemeriksaan ini dapat dilakukan terhadap konfigurasi yang tidak memenuhi prinsip keamanan atau praktik terbaik, atau perubahan yang dibuat antara konfigurasi yang diuji dan yang di-deploy. Untuk aplikasi yang berjalan, Anda dapat memeriksa apakah konfigurasi telah diubah dengan cara yang tidak diharapkan, termasuk yang di luar peristiwa penskalaan otomatis atau deployment yang tidak dikenal.

Untuk deteksi bagian yang kedua, perilaku yang tidak diharapkan, Anda dapat menggunakan alat atau memberikan peringatan saat terjadi peningkatan jenis panggilan API tertentu. Menggunakan Amazon GuardDuty, Anda dapat selalu menerima peringatan saat terdapat aktivitas yang tidak diharapkan dan berpotensi tidak sah atau berbahaya di akun AWS Anda. Anda juga harus memantau secara langsung perubahan panggilan API yang tidak Anda maksudkan untuk digunakan dalam beban kerja Anda, serta panggilan API yang mengubah postur keamanan.

Deteksi memungkinkan Anda untuk mengidentifikasi potensi kesalahan konfigurasi keamanan, ancaman, atau perilaku yang tidak diharapkan. Ini merupakan bagian yang sangat penting dalam siklus hidup keamanan dan dapat digunakan untuk mendukung proses yang berkualitas, kewajiban kepatuhan atau hukum, serta upaya identifikasi dan respons terhadap ancaman. Ada beberapa jenis mekanisme deteksi. Misalnya, log dari beban kerja Anda dapat dianalisis untuk exploit yang digunakan. Anda harus meninjau secara rutin mekanisme deteksi yang terkait dengan beban kerja Anda guna memastikan bahwa Anda telah memenuhi persyaratan dan kebijakan internal serta eksternal. Notifikasi dan peringatan otomatis harus didasarkan pada kondisi yang telah ditetapkan agar tim atau alat Anda dapat melakukan penyelidikan. Mekanisme-mekanisme ini merupakan faktor reaktif penting yang dapat membantu organisasi Anda mengidentifikasi dan memahami cakupan aktivitas anomali.

Di AWS, ada beberapa pendekatan yang dapat Anda gunakan saat menangani mekanisme deteksi. Bagian berikut akan menjelaskan cara menggunakan pendekatan ini:

Praktik terbaik

SEC04-BP01 Mengonfigurasi pencatatan log layanan dan aplikasi

Menyimpan log peristiwa keamanan dari layanan dan aplikasi. Hal ini merupakan prinsip fundamental dalam keamanan untuk audit, penyelidikan, dan kasus penggunaan operasional, serta merupakan persyaratan keamanan umum yang didorong oleh prosedur, kebijakan, dan standar tata kelola, risiko, serta kepatuhan (GRC).

Hasil yang diinginkan: Sebuah organisasi harus dapat dengan andal dan konsisten mengambil log peristiwa keamanan dari aplikasi dan layanan AWS dengan cepat saat diperlukan untuk memenuhi proses atau kewajiban internal, seperti respons insiden keamanan. Sebaiknya pusatkan log untuk mendapatkan hasil operasional yang lebih baik.

Antipola umum:

- Log disimpan tanpa batas waktu yang jelas atau dihapus terlalu cepat.
- Semua orang dapat mengakses log.
- Sepenuhnya menggunakan proses manual untuk tata kelola dan penggunaan log.
- Menyimpan setiap jenis log untuk berjaga-jaga jika diperlukan.
- Memeriksa integritas log hanya jika diperlukan.

Manfaat menjalankan praktik terbaik ini: Mengimplementasikan mekanisme analisis akar masalah (RCA) untuk insiden keamanan dan sumber bukti untuk kewajiban tata kelola, risiko, dan kepatuhan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Selama penyelidikan keamanan atau kasus penggunaan lain berdasarkan kebutuhan Anda, Anda harus dapat meninjau log yang relevan untuk mencatat dan memahami seluruh cakupan serta garis waktu insiden. Log juga diperlukan untuk pembuatan peringatan yang mengindikasikan bahwa

tindakan tertentu telah terjadi. Sangat penting untuk memilih, mengaktifkan, menyimpan, dan menyiapkan mekanisme kueri dan pengambilan serta pembuatan peringatan.

Langkah implementasi

- Pilih dan aktifkan sumber log. Sebelum melakukan penyelidikan keamanan, Anda perlu mengambil log yang relevan untuk merekonstruksi aktivitas secara surut di Akun AWS. Pilih dan aktifkan sumber log yang relevan dengan beban kerja Anda.

  Kriteria pemilihan sumber log harus didasarkan pada kasus penggunaan yang diperlukan oleh bisnis Anda. Tetapkan jejak untuk setiap Akun AWS menggunakan AWS CloudTrail atau jejak AWS Organizations, dan konfigurasikan bucket Amazon S3 untuk jejak tersebut.

  AWS CloudTrail adalah layanan pencatatan log yang melacak panggilan API yang dibuat terhadap Akun AWS yang merekam aktivitas layanan AWS. Layanan tersebut diaktifkan secara default dengan peristiwa manajemen retensi 90 hari yang dapat diambil melalui riwayat Peristiwa CloudTrail menggunakan AWS Management Console, AWS CLI, atau AWS SDK. Untuk visibilitas dan retensi peristiwa data yang lebih lama, buat jejak CloudTrail dan kaitkan dengan bucket Amazon S3, serta dengan grup log Amazon CloudWatch (opsional). Anda juga dapat membuat CloudTrail Lake, yang menyimpan log CloudTrail hingga tujuh tahun dan menyediakan fasilitas kueri berbasis SQL.

  AWS menyarankan agar pelanggan yang menggunakan VPC mengaktifkan lalu lintas jaringan dan log DNS menggunakan Log Arus VPC dan log kueri Amazon Route 53 Resolver, serta mengalirkannya ke bucket Amazon S3 atau grup log CloudWatch. Anda dapat membuat log arus VPC untuk VPC, subnet, dan antarmuka jaringan. Untuk Log Arus VPC, Anda dapat memilih cara dan tempat penggunaan Log Arus untuk mengurangi biaya.

  Log AWS CloudTrail, Log Arus VPC, dan log kueri Route 53 Resolver merupakan sumber pencatatan log dasar untuk mendukung penyelidikan keamanan di AWS. Anda juga dapat menggunakan Danau Keamanan Amazon untuk mengumpulkan, menormalisasi, dan menyimpan data log ini dalam format Apache Parquet dan Open Cybersecurity Schema Framework (OCSF), yang siap untuk kueri. Danau Keamanan juga mendukung log AWS lainnya dan log dari sumber pihak ketiga.

  Layanan AWS dapat membuat log yang tidak direkam oleh sumber log dasar, seperti log Elastic Load Balancing, log AWS WAF, log perekam AWS Config, temuan Amazon GuardDuty, log audit Amazon Elastic Kubernetes Service (Amazon EKS), dan log aplikasi serta sistem operasi instans Amazon EC2. Untuk daftar lengkap opsi pencatatan log dan pemantauan, lihat Lampiran A:

Penentuan kemampuan cloud – Pencatatan Log dan Peristiwa dalam Panduan Respons Insiden Keamanan AWS.

- Pelajari kemampuan pencatatan log untuk setiap aplikasi dan layanan AWS: Setiap layanan dan aplikasi AWS memberikan opsi untuk penyimpanan log yang masing-masing dilengkapi dengan kemampuan retensi dan siklus hidup. Dua layanan penyimpanan yang paling umum adalah Amazon Simple Storage Service (Amazon S3) dan Amazon CloudWatch. Untuk periode retensi yang panjang, sebaiknya gunakan Amazon S3 untuk efektivitas biaya dan kemampuan siklus hidup yang fleksibel. Jika opsi pencatatan log utama adalah Log Amazon CloudWatch, sebagai opsi, Anda dapat mempertimbangkan untuk mengarsipkan log yang jarang diakses ke Amazon S3.
- Pilih penyimpanan log: Pilihan penyimpanan log umumnya dikaitkan dengan alat kueri yang digunakan, kemampuan retensi, seberapa familier, dan biaya. Opsi utama untuk penyimpanan log adalah bucket Amazon S3 atau grup Log CloudWatch.

  Bucket Amazon S3 menyediakan penyimpanan yang tahan lama dan hemat biaya, dengan kebijakan siklus hidup opsional. Log yang disimpan di bucket Amazon S3 dapat dikueri menggunakan layanan seperti Amazon Athena.

  Grup log CloudWatch menyediakan penyimpanan yang tahan lama dan fasilitas kueri bawaan melalui Wawasan Log CloudWatch.
- Identifikasi retensi log yang sesuai: Saat Anda menggunakan bucket Amazon S3 atau grup log CloudWatch untuk menyimpan log, Anda harus menetapkan siklus hidup yang memadai untuk setiap sumber log guna mengoptimalkan biaya penyimpanan dan pengambilan. Pada umumnya, pelanggan memiliki waktu antara tiga bulan hingga satu tahun untuk melakukan kueri log, dengan periode retensi hingga tujuh tahun. Pilihan ketersediaan dan retensi harus selaras dengan persyaratan keamanan dan gabungan dari undang-undang, peraturan, serta kewajiban bisnis.
- Aktifkan pencatatan log untuk setiap layanan dan aplikasi AWS dengan kebijakan retensi dan siklus hidup yang tepat: Untuk setiap aplikasi dan layanan AWS di organisasi Anda, cari panduan konfigurasi pencatatan log khusus:
  - Mengonfigurasi Jejak AWS CloudTrail
  - Mengonfigurasi Log Arus VPC
  - Mengonfigurasi Ekspor Temuan Amazon GuardDuty
  - Mengonfigurasi perekaman AWS Config
  - Mengonfigurasi lalu lintas ACL web AWS WAF
  - Mengonfigurasi log lalu lintas jaringan AWS Network Firewall

  - Mengonfigurasi log akses Elastic Load Balancing
  - Mengonfigurasi log kueri Amazon Route 53 Resolver
  - Mengonfigurasi log Amazon RDS
  - Mengonfigurasi log Bidang Kendali Amazon EKS
  - Mengonfigurasi agen Amazon CloudWatch untuk instans Amazon EC2 dan server on-premise
- Pilih dan implementasikan mekanisme kueri untuk log: Untuk kueri log, Anda dapat menggunakan CloudWatch Wawasan Log untuk data yang disimpan di grup log CloudWatch, dan Amazon Athena serta Amazon OpenSearch Service untuk data yang disimpan di Amazon S3. Anda dapat menggunakan alat kueri pihak ketiga seperti layanan informasi keamanan dan manajemen peristiwa (SIEM).

  Proses pemilihan alat kueri harus mempertimbangkan aspek manusia, proses, dan teknologi operasi keamanan Anda. Pilih alat yang memenuhi persyaratan operasional, bisnis, dan keamanan, serta dapat diakses dan dipelihara dalam jangka panjang. Perlu diingat bahwa alat kueri berfungsi secara optimal saat jumlah log yang dipindai masih berada dalam batasan alat tersebut. Tidak jarang terdapat beberapa alat kueri karena adanya kendala biaya atau teknis.

  Misalnya, Anda menggunakan informasi keamanan dan manajemen peristiwa pihak ketiga untuk menjalankan kueri data selama 90 hari terakhir, tetapi menggunakan Athena untuk menjalankan kueri di atas 90 hari karena biaya penyerapan log SIEM. Terlepas dari implementasi, pastikan pendekatan Anda meminimalkan jumlah alat yang diperlukan untuk memaksimalkan efisiensi operasional, khususnya selama penyelidikan peristiwa keamanan.
- Gunakan log untuk peringatan: AWS memberikan peringatan melalui beberapa layanan keamanan:
  - AWS Config memantau dan merekam konfigurasi sumber daya AWS Anda serta membantu mengotomatiskan evaluasi dan perbaikan berdasarkan konfigurasi yang diinginkan.
  - Amazon GuardDuty adalah layanan deteksi ancaman yang terus memantau aktivitas mencurigakan dan perilaku tidak sah untuk melindungi Akun AWS dan beban kerja Anda. GuardDuty menyerap, menggabungkan, dan menganalisis informasi dari berbagai sumber, seperti manajemen dan peristiwa data AWS CloudTrail, log DNS, Log Arus VPC, dan log Audit Amazon EKS. GuardDuty mengambil aliran data independen secara langsung dari CloudTrail, Log Arus VPC, log kueri DNS, dan Amazon EKS. Anda tidak perlu mengelola kebijakan bucket Amazon S3 atau mengubah cara Anda mengumpulkan dan menyimpan log. Sebaiknya tetap simpan log tersebut untuk tujuan penyelidikan dan kepatuhan.
  - AWS Security Hub menyediakan satu tempat yang mengumpulkan, mengatur, dan memprioritaskan peringatan keamanan atau temuan Anda dari beberapa layanan AWS serta

produk pihak ketiga opsional untuk menampilkan peringatan keamanan dan status kepatuhan secara komprehensif.

Anda juga dapat menggunakan mesin pembuat peringatan kustom untuk peringatan keamanan yang tidak dicakup oleh layanan ini atau untuk peringatan tertentu yang relevan dengan lingkungan Anda. Untuk informasi tentang pembuatan peringatan dan deteksi tersebut, lihat Deteksi dalam Panduan Respons Insiden Keamanan AWS.

Sumber daya

Praktik Terbaik Terkait:

- SEC04-BP02 Menganalisis log, temuan, dan metrik secara terpusat
- SEC07-BP04 Menentukan manajemen siklus hidup data
- SEC10-BP06 Melakukan deployment alat di awal

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS
- Memulai Danau Keamanan Amazon
- Memulai: Amazon CloudWatch Logs
- Solusi Partner Keamanan: Pencatatan Log dan Pemantauan

Video terkait:

- AWS re:Invent 2022 - Memperkenalkan Danau Keamanan Amazon

Contoh terkait:

- Assisted Log Enabler untuk AWS
- Ekspor Temuan AWS Security Hub Historis

Alat terkait:

- Snowflake for Cybersecurity

SEC04-BP02 Menganalisis log, temuan, dan metrik secara terpusat

Tim operasi keamanan mengandalkan pengumpulan data dan penggunaan alat pencarian untuk menemukan potensi peristiwa yang menjadi perhatian, yang mungkin menandakan aktivitas yang tidak diotorisasi atau perubahan yang tidak diinginkan. Namun, menganalisis data yang terkumpul dan memproses informasi secara manual saja tidak cukup untuk mengimbangi volume informasi yang mengalir dari arsitektur kompleks. Analisis dan pelaporan saja belum cukup untuk memfasilitasi penetapan sumber daya yang tepat untuk mengerjakan peristiwa pada waktu yang diinginkan.

Praktik terbaik untuk membangun tim operasi keamanan yang matang adalah dengan mengintegrasikan aliran peristiwa keamanan dan temuan ke dalam notifikasi dan sistem alur kerja seperti sistem ticketing, sistem masalah atau bug, atau sistem informasi keamanan dan manajemen peristiwa (SIEM) lainnya. Hal ini mengalihkan alur kerja dari email dan laporan statis, sehingga Anda dapat merutekan, mengeskalasi, dan mengelola peristiwa atau temuan. Banyak organisasi yang juga mengintegrasikan peringatan keamanan ke dalam obrolan atau kolaborasi mereka, dan platform produktivitas developer. Untuk organisasi yang memulai otomatisasi, sistem ticketing yang didorong API dan berlatensi rendah menawarkan berbagai fleksibilitas saat merencanakan apa yang harus diotomatiskan terlebih dahulu.

Praktik terbaik ini tidak hanya berlaku untuk peristiwa keamanan yang dibuat dari pesan log yang menggambarkan aktivitas pengguna atau peristiwa jaringan, tetapi juga dari perubahan yang terdeteksi dalam infrastruktur. Kemampuan untuk mendeteksi perubahan, menentukan apakah perubahan tersebut sesuai, dan kemudian mengarahkan informasi tersebut ke alur kerja remediasi begitu penting dalam memelihara dan memvalidasi arsitektur yang aman, saat terjadi perubahan yang tidak diinginkan tetapi sulit dideteksi sehingga eksekusinya saat ini tidak dapat dicegah dengan kombinasi konfigurasi AWS Identity and Access Management(IAM) dan AWS Organizations.

Amazon GuardDuty dan AWS Security Hub memberikan gabungan, deduplikasi, dan mekanisme analisis untuk catatan log, yang juga dibuat tersedia untuk Anda via layanan AWS lainnya. GuardDuty menyerap, menggabungkan, dan menganalisis, informasi dari sumber seperti manajemen AWS CloudTrail dan peristiwa data, log DNS VPC, dan Log Alur VPC. Security Hub dapat menyerap, menggabungkan, dan menganalisis hasil dari GuardDuty, AWS Config, Amazon Inspector, Amazon Macie, AWS Firewall Manager, dan sebagian besar produk keamanan pihak ketiga yang tersedia di AWS Marketplace, dan jika langsung dibangun, kode milik Anda sendiri. GuardDuty dan Security Hub memiliki model Administrator-Anggota yang dapat mengagregatkan temuan dan wawasan dari beberapa akun, dan Security Hub sering digunakan oleh pelanggan SIEM on-premise sebagai log sisi AWS dan peringatan praprosesor dan agregator yang dapat diserap Amazon EventBridge melalui prosesor dan penerus berbasis AWS Lambda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Tinggi

Panduan implementasi

- Evaluasikan kemampuan pemrosesan log: Evaluasikan opsi yang tersedia untuk pemrosesan log.
  - Gunakan Amazon OpenSearch Service untuk mencatat dan memantau (hampir) semuanya
  - Temukan partner yang ahli dalam pencatatan log dan pemantauan.
- Sebagai awal untuk menganalisis log CloudTrail, uji Amazon Athena.
  - Konfigurasikan Athena untuk menganalisis log CloudTrail
- Implementasikan sentralisasi pencatatan dalam AWS: Lihat solusi contoh AWS berikut untuk memusatkan pencatatan dari berbagai sumber.
  - Solusi sentralisasi pencatatan
- Implementasikan sentralisasi pencatatan dengan partner: APN Partner memiliki solusi untuk membantu Anda menganalisis log secara terpusat.
  - Pencatatan dan Pemantauan

Sumber daya

Dokumen terkait:

- AWS Answers: Pencatatan Log Terpusat
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- Mulai menggunakan: Amazon CloudWatch Logs
- Solusi Partner Keamanan: Pencatatan Log dan Pemantauan

Video terkait:

- Konfigurasi dan Kepatuhan Sumber Daya Pemantauan Terpusat
- Memperbaiki Temuan Amazon GuardDuty dan AWS Security Hub
- Manajemen ancaman di cloud: Amazon GuardDuty dan AWS Security Hub

SEC04-BP03 Mengotomatiskan respons untuk peristiwa

Menggunakan otomatisasi untuk menyelidiki dan menangani peristiwa dapat mengurangi upaya manusia dan potensi kesalahan, serta memungkinkan Anda untuk menskalakan kemampuan penyelidikan. Tinjauan rutin dapat membantu Anda menyesuaikan alat otomatisasi, dan mengulanginya secara iteratif.

Di AWS, menyelidiki peristiwa menarik dan informasi tentang potensi perubahan yang tidak diharapkan ke alur kerja otomatis dapat dicapai menggunakan Amazon EventBridge. Layanan ini menyediakan mesin aturan yang dapat diskalakan dan dirancang untuk mengelola format peristiwa AWS native (seperti peristiwa AWS CloudTrail), serta peristiwa kustom yang dapat dihasilkan dari aplikasi Anda. Amazon GuardDuty juga memungkinkan Anda untuk merutekan peristiwa ke sistem alur kerja untuk mereka yang membangun sistem respons insiden (AWS Step Functions), atau ke Akun Keamanan pusat, atau ke bucket untuk analisis lebih jauh.

Mendeteksi perubahan dan merutekan informasi ini ke alur kerja yang sesuai dapat dilakukan dengan menggunakan Aturan AWS Config dan Paket Konformasi. AWS Config mendeteksi perubahan ke layanan dalam cakupan (walaupun dengan latensi yang lebih tinggi dari EventBridge) dan membuat peristiwa yang dapat di-parse menggunakan Aturan AWS Config untuk rollback, penerapan kebijakan kepatuhan, dan melanjutkan informasi ke sistem, seperti mengubah platform manajemen dan sistem ticketing operasional. Selain menulis fungsi Lambda Anda sendiri untuk merespons peristiwa AWS Config, Anda juga dapat memanfaatkan Kit Pengembangan Aturan AWS Config, dan pustaka sumber terbuka Aturan AWS Config. Paket konformasi adalah kumpulan Aturan AWS Config dan tindakan perbaikan yang Anda deploy sebagai entitas tunggal yang ditulis sebagai templat YAML. Sebuah sampel templat paket konformasi tersedia untuk Pilar Keamanan Well-Architected.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

- Implementasikan peringatan yang diotomatiskan dengan GuardDuty: GuardDuty adalah layanan deteksi ancaman yang terus-menerus memantau aktivitas berbahaya dan perilaku yang tidak diotorisasi, untuk melindungi Akun AWS dan beban kerja Anda. Aktifkan GuardDuty dan konfigurasikan peringatan yang diotomatiskan.
- Otomatiskan proses penyelidikan: Kembangkan proses otomatis yang menyelidiki peristiwa serta melaporkan informasi ke administrator untuk menghemat waktu.
  - Lab: Penggunaan Amazon GuardDuty

Sumber daya

Dokumen terkait:

- AWS Jawaban: Pencatatan Log Terpusat
- AWS Security Hub
- Amazon CloudWatch
- Amazon EventBridge
- Mulai menggunakan: Amazon CloudWatch Logs
- Solusi Partner Keamanan: Pencatatan Log dan Pemantauan
- Menyiapkan Amazon GuardDuty

Video terkait:

- Centrally Monitoring Resource Configuration and Compliance
- Memperbaiki Temuan Amazon GuardDuty dan AWS Security Hub
- Manajemen ancaman di cloud: Amazon GuardDuty dan AWS Security Hub

Contoh terkait:

- Lab: Deployment Otomatis Kontrol Deteksi

SEC04-BP04 Implementasikan peristiwa keamanan yang dapat ditindaklanjuti

Buat peringatan yang dikirimkan ke tim Anda dan dapat ditindaklanjuti oleh mereka. Pastikan peringatan mencakup informasi yang relevan bagi tim untuk mengambil tindakan. Untuk setiap mekanisme deteksi yang Anda miliki, Anda juga harus memiliki proses, dalam bentuk runbook atau playbook, untuk menyelidiki. Contohnya, ketika Anda mengaktifkan Amazon GuardDuty, ini menghasilkan temuan yang berbeda.. Anda harus memiliki entri runbook untuk setiap jenis temuan, contohnya, jika ditemukan trojan, runbook Anda memiliki instruksi mudah yang menginstruksikan seseorang untuk menyelidiki dan memperbaiki.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Temukan metrik yang tersedia untuk layanan AWS: Temukan metrik yang tersedia melalui Amazon CloudWatch untuk layanan yang Anda gunakan.
  - Dokumentasi layanan AWS
  - Menggunakan Metrik Amazon CloudWatch
- Konfigurasikan alarm Amazon CloudWatch.
  - Menggunakan Alarm Amazon CloudWatch

Sumber daya

Dokumen terkait:

- Amazon CloudWatch
- Amazon EventBridge
- Solusi Partner Keamanan: Logging dan Pemantauan

Video terkait:

- Konfigurasi dan Kepatuhan Sumber Daya Pemantauan Terpusat
- Memperbaiki Temuan Amazon GuardDuty dan AWS Security Hub
- Manajemen ancaman di cloud: Amazon GuardDuty dan AWS Security Hub

Perlindungan infrastruktur

Perlindungan infrastruktur berkenaan dengan metodologi kontrol, seperti pertahanan mendalam, yang diperlukan untuk memenuhi praktik terbaik dan kewajiban organisasi atau peraturan. Penggunaan metodologi ini vital untuk keberhasilan dan keberlangsungan operasi di cloud.

Perlindungan infrastruktur adalah bagian penting di sebuah program keamanan informasi. Fungsinya adalah untuk memastikan sistem dan layanan dalam beban kerja Anda terlindungi dari potensi kelemahan serta akses yang tidak diinginkan dan tidak sah. Sebagai contoh, Anda akan menetapkan batasan kepercayaan (misalnya batasan jaringan dan akun), konfigurasi dan pemeliharaan keamanan sistem (misalnya penguatan, perampingan, dan penambalan), autentikasi dan otorisasi sistem operasi (misalnya pengguna, kunci, dan tingkat akses), dan titik-titik penegakan kebijakan yang tepat lainnya (misalnya firewall aplikasi dan/atau gateway API).

Wilayah, Zona Ketersediaan, AWS Local Zones, dan AWS Outposts

Pastikan Anda familer dengan konsep Wilayah, Zona Ketersediaan, AWS Local Zones, dan AWS Outposts, yakni komponen infrastruktur global aman AWS.

AWS memiliki konsep Wilayah, yakni lokasi fisik di seluruh dunia tempat kami mengklaster pusat-pusat data. Kami menyebut setiap kelompok pusat data dengan sebutan Zona Ketersediaan (AZ). Setiap Wilayah AWS terdiri dari beberapa AZ yang terisolasi dan terpisah secara fisik di sebuah area geografis. Jika Anda memiliki persyaratan residensi data, Anda dapat memilih Wilayah AWS yang dekat dengan lokasi yang Anda inginkan. Anda memegang penuh kontrol dan kepemilikan atas Wilayah tempat data Anda disimpan secara fisik, dan ini bermanfaat untuk memenuhi persyaratan kepatuhan wilayah dan residensi data Anda. Masing-masing AZ memiliki daya, pendingin, dan keamanan fisik independen. Jika suatu aplikasi dipartisi lintas AZ, Anda akan lebih terisolasi dan terlindungi dari permasalahan seperti pemadaman listrik, sambaran petir, angin topan, gempa bumi, dan lain-lain. AZ secara fisik terpisah dengan jarak yang cukup jauh, berkilo-kilo meter, dari AZ lain, meskipun semuanya berada dalam jarak 100 km (60 mil) dari satu sama lain. Semua AZ di sebuah Wilayah AWS saling terhubung dengan bandwidth tinggi, jaringan latensi rendah, menggunakan serat metro khusus yang sepenuhnya redundan, yang menyediakan jaringan throughput tinggi dan latensi rendah antara AZ. Semua lalu lintas antara AZ dienkripsi. Pelanggan AWS yang berfokus pada ketersediaan tinggi dapat merancang aplikasi mereka agar berjalan di beberapa AZ guna mencapai toleransi kesalahan yang jauh lebih besar. Wilayah AWS memenuhi tingkat keamanan, kepatuhan, dan perlindungan data tertinggi.

Zona Lokal AWS menempatkan layanan komputasi, penyimpanan, basis data, dan layanan AWS terpilih lainnya lebih dekat dengan pengguna akhir. Dengan Zona Lokal AWS, Anda dapat dengan mudah menjalankan aplikasi sangat berat yang memerlukan latensi satu digit milidetik ke pengguna akhir, seperti pembuatan media dan konten hiburan, gaming waktu nyata, simulasi penampungan air, otomatisasi desain elektronik, dan machine learning. Setiap lokasi Zona Lokal AWS adalah perluasan Wilayah AWS di mana Anda dapat menjalankan aplikasi peka latensi, menggunakan layanan AWS seperti Amazon EC2, Amazon VPC, Amazon EBS, Amazon File Storage, dan Elastic Load Balancing di lokasi geografis yang dekat dengan pengguna akhir. Zona Lokal AWS menyediakan bandwidth tinggi, koneksi aman antara beban kerja lokal, dan yang berjalan di Wilayah AWS, sehingga Anda dapat terhubung secara lancar ke berbagai layanan dalam wilayah melalui API dan set alat yang sama.

AWS Outposts menghadirkan layanan, infrastruktur, dan model operasi AWS native ke semua pusat data, ruang lokasi bersama, atau fasilitas on-premise. Anda dapat menggunakan API, alat, dan infrastruktur AWS di semua fasilitas on-premise dan AWS Cloud untuk menghadirkan pengalaman hybrid yang benar-benar konsisten. AWS Outposts dirancang untuk lingkungan terkoneksi dan dapat digunakan untuk mendukung beban kerja yang harus tetap on-premise dikarenakan kebutuhan latensi yang rendah atau pemrosesan data lokal.

Di AWS, terdapat sejumlah pendekatan perlindungan infrastruktur. Bagian berikutnya akan menjelaskan cara menggunakan pendekatan ini.

Topik

- Melindungi jaringan
- Melindungi komputasi

Melindungi jaringan

Pengguna, baik dari tenaga kerja maupun pelanggan Anda, bisa berada di mana saja. Anda perlu beralih dari model tradisional yang memercayai semua orang dan semua hal yang memiliki akses ke jaringan Anda. Ketika Anda mengikuti prinsip penerapan keamanan di semua lapisan, berarti Anda menerapkan pendekatan Zero Trust. Keamanan Zero Trust adalah model di mana komponen aplikasi atau layanan mikro dianggap terpisah satu sama lain dan komponen atau layanan mikro tersebut tidak percaya satu sama lain.

Perencanaan dan manajemen yang cermat pada desain jaringan Anda membentuk fondasi bagi Anda untuk menyediakan pemisahan dan batasan untuk sumber daya di dalam beban kerja.

Karena banyak sumber daya di beban kerja Anda beroperasi di VPC dan mewarisi properti keamanan, desain sangat perlu untuk didukung dengan mekanisme penyelidikan dan perlindungan yang diperkuat oleh otomatisasi. Demikian juga, untuk beban kerja yang beroperasi di luar VPC, menggunakan layanan murni edge dan/atau nirserver, praktik terbaik tersebut berlaku dalam pendekatan yang lebih sederhana. Lihat Lensa Aplikasi Nirserver AWS Well-Architected untuk mendapatkan panduan khusus tentang keamanan nirserver.

Praktik terbaik

- SEC05-BP01 Membuat lapisan jaringan
- SEC05-BP02 Mengontrol lalu lintas di semua lapisan
- SEC05-BP03 Mengotomatiskan perlindungan jaringan
- SEC05-BP04 Mengimplementasikan inspeksi dan perlindungan

SEC05-BP01 Membuat lapisan jaringan

Kelompokkan komponen dengan persyaratan sensitivitas yang sama ke dalam lapisan-lapisan untuk meminimalkan cakupan potensi dampak dari akses yang tidak sah. Misalnya, sebuah klaster basis data di cloud privat virtual (VPC) yang tidak memerlukan akses internet harus ditempatkan dalam subnet yang tidak memiliki rute ke atau dari internet. Lalu lintas hanya boleh mengalir dari sumber daya terdekat berikutnya yang paling tidak sensitif. Pertimbangkan aplikasi web di balik penyeimbang beban. Basis data Anda tidak boleh dapat diakses secara langsung dari penyeimbang beban. Hanya logika bisnis dan server web yang boleh memiliki akses langsung ke basis data Anda.

Hasil yang diinginkan: Membuat jaringan berlapis. Jaringan berlapis membantu mengelompokkan komponen jaringan yang serupa secara logis. Jaringan ini memperkecil potensi cakupan dampak dari akses jaringan yang tidak sah. Jaringan berlapis yang tepat mempersulit pengguna yang tidak sah untuk beralih ke sumber daya tambahan di lingkungan AWS Anda. Selain mengamankan jalur jaringan internal, Anda juga perlu melindungi edge jaringan, seperti aplikasi web dan titik akhir API.

Antipola umum:

- Membuat semua sumber daya dalam satu VPC atau subnet.
- Menggunakan grup keamanan yang terlalu permisif.
- Gagal menggunakan subnet.
- Mengizinkan akses langsung ke penyimpanan data seperti basis data.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Komponen seperti instans Amazon Elastic Compute Cloud (Amazon EC2), klaster basis data Amazon Relational Database Service (Amazon RDS), dan fungsi AWS Lambda yang memiliki persyaratan keterjangkauan yang sama dapat disegmentasikan menjadi lapisan yang dibentuk oleh subnet. Pertimbangkan untuk melakukan deployment beban kerja nirserver, seperti fungsi Lambda, dalam VPC atau di balik Amazon API Gateway. Tugas AWS Fargate (Fargate) yang tidak memerlukan akses internet harus ditempatkan di subnet yang tidak memiliki rute ke atau dari internet. Pendekatan berlapis ini memitigasi dampak kesalahan konfigurasi satu lapisan yang dapat mengizinkan akses yang tidak diinginkan. Untuk AWS Lambda, Anda dapat menjalankan fungsi di VPC untuk memanfaatkan kontrol berbasis VPC.

Untuk konektivitas jaringan yang dapat mencakup ribuan VPC, Akun AWS, dan jaringan on-premise, Anda harus menggunakan AWS Transit Gateway. Transit Gateway bertindak sebagai hub yang mengontrol cara perutean lalu lintas di antara semua jaringan yang terhubung, yang bertindak seperti jari-jari roda. Lalu lintas antara Amazon Virtual Private Cloud (Amazon VPC) dan Transit Gateway tetap berada dalam jaringan privat AWS, sehingga mengurangi paparan eksternal terkait pengguna yang tidak sah dan potensi masalah keamanan. Peering Antarwilayah Transit Gateway juga mengenkripsi lalu lintas Antarwilayah tanpa titik kegagalan tunggal atau hambatan bandwidth.

Langkah implementasi

- Gunakan Reachability Analyzer untuk menganalisis jalur antara sumber dan tujuan berdasarkan konfigurasi: Reachability Analyzer memungkinkan Anda untuk mengotomatiskan verifikasi konektivitas ke dan dari sumber daya yang terhubung ke VPC. Perhatikan bahwa analisis ini dilakukan dengan meninjau konfigurasi (tidak ada paket jaringan yang dikirimkan dalam menjalankan analisis ini).
- Gunakan Penganalisis Akses Jaringan Amazon VPC untuk mengidentifikasi akses jaringan yang tidak diinginkan ke sumber daya: Penganalisis Akses Jaringan Amazon VPC memungkinkan Anda untuk menentukan persyaratan akses jaringan dan mengidentifikasi jalur jaringan potensial.
- Pertimbangkan apakah sumber daya harus ada di subnet publik: Jangan menempatkan sumber daya di subnet publik VPC Anda kecuali sumber daya benar-benar harus menerima lalu lintas jaringan masuk dari sumber publik.
- Buat subnet di VPC Anda: Buat subnet untuk setiap lapisan jaringan (dalam grup yang menyertakan beberapa Zona Ketersediaan) untuk meningkatkan segmentasi mikro. Selain itu,

pastikan Anda telah mengaitkan tabel rute yang benar ke subnet Anda untuk mengontrol perutean dan konektivitas internet.

- Gunakan AWS Firewall Manager untuk mengelola grup keamanan VPC Anda: AWS Firewall Manager membantu mengurangi beban manajemen dalam penggunaan beberapa grup keamanan.
- Gunakan AWS WAF untuk melindungi dari kerentanan web umum: AWS WAF dapat membantu meningkatkan keamanan edge dengan memeriksa lalu lintas untuk kerentanan web umum, misalnya injeksi SQL. Layanan ini juga dapat Anda gunakan untuk membatasi alamat IP yang berasal dari negara atau lokasi geografis tertentu.
- Gunakan Amazon CloudFront sebagai jaringan distribusi konten (CDN): Amazon CloudFront dapat membantu mempercepat aplikasi web dengan menyimpan data lebih dekat ke pengguna. Hal ini juga meningkatkan keamanan edge dengan menerapkan HTTPS, membatasi akses ke area geografis, dan memastikan bahwa lalu lintas jaringan hanya dapat mengakses sumber daya saat dirutekan melalui CloudFront.
- Gunakan Amazon API Gateway saat membuat antarmuka pemrograman aplikasi (API): Amazon API Gateway membantu menerbitkan, memantau, dan mengamankan API WebSocket, HTTPS, dan REST.

Sumber daya

Dokumen terkait:

- AWS Firewall Manager
- Amazon Inspector
- Keamanan Amazon VPC
- Reachability Analyzer
- Penganalisis Akses Jaringan Amazon VPC

Video terkait:

- Arsitektur referensi AWS Transit Gateway untuk banyak VPC
- Perlindungan dan Akselerasi Aplikasi dengan Amazon CloudFront, AWS WAF, dan AWS Shield
- AWS re:Inforce 2022 - Validasikan kontrol akses jaringan efektif di AWS
- AWS re:Inforce 2022 - Perlindungan tingkat lanjut terhadap bot menggunakan AWS WAF

Contoh terkait:

- Well-Architected Lab - Deployment Otomatis VPC
- Lokakarya: Penganalisis Akses Jaringan Amazon VPC

SEC05-BP02 Mengontrol lalu lintas di semua lapisan

Ketika merancang topologi jaringan, Anda harus memeriksa persyaratan konektivitas setiap komponen. Misalnya, periksa apakah komponen memerlukan aksesibilitas internet (masuk dan keluar), konektivitas ke VPC, layanan edge, dan pusat data eksternal.

Dengan VPC, Anda dapat menentukan topologi jaringan yang menjangkau Wilayah AWS dengan rentang alamat IPv4 privat yang Anda atur, atau rentang alamat IPv6 yang dipilih oleh AWS. Anda harus menerapkan beberapa kontrol dengan pendekatan pertahanan mendalam untuk lalu lintas masuk dan keluar, termasuk penggunaan grup keamanan (firewall inspeksi stateful), ACL Jaringan, subnet, dan tabel rute. Di dalam VPC, Anda dapat membuat subnet di Zona Ketersediaan. Masing-masing subnet dapat memiliki tabel rute terkait yang menentukan aturan perutean untuk mengelola jalur yang digunakan lalu lintas dalam subnet. Anda dapat menentukan subnet yang dapat dirutekan internet dengan rute yang mengarah ke gateway NAT atau internet yang terikat ke VPC, atau melalui VPC lainnya.

Saat diluncurkan di dalam VPC, instans, basis data Amazon Relational Database Service(Amazon RDS), atau layanan lainnya akan memiliki grup keamanannya sendiri per antarmuka jaringan. Firewall ini berada di luar lapisan sistem operasi dan dapat digunakan untuk menentukan aturan bagi lalu lintas masuk dan keluar yang diizinkan. Anda juga dapat menentukan hubungan antargrup keamanan. Misalnya, instans dalam grup keamanan tingkat basis data hanya menerima lalu lintas dari instans dalam tingkat aplikasi, dengan merujuk ke grup keamanan yang diterapkan ke instans yang terlibat. Namun jika Anda menggunakan protokol non-TCP, Anda tidak perlu memiliki instans Amazon Elastic Compute Cloud(Amazon EC2) yang dapat diakses langsung dengan internet (bahkan dengan port yang dibatasi oleh grup keamanan) tanpa penyeimbang beban, atau CloudFront. Hal ini akan membantu melindunginya dari akses yang tidak diharapkan melalui sistem operasi atau masalah aplikasi. Subnet juga dapat memiliki ACL jaringan yang terikat dengannya, yang berperan sebagai firewall stateless. Anda harus mengonfigurasi jaringan ACL untuk mempersempit cakupan lalu lintas yang diizinkan antarlapisan, Anda juga harus menentukan aturan masuk dan keluar.

Beberapa layanan AWS memerlukan komponen untuk mengakses internet guna membuat panggilan API, tempat titik akhir API AWS berada. Layanan AWS lain menggunakan titik akhir VPC di dalam

Amazon VPC Anda. Banyak layanan AWS, termasuk Amazon S3 dan Amazon DynamoDB, yang mendukung titik akhir VPC. Selain itu, teknologi ini telah digeneralisasi dalam AWS PrivateLink. Sebaiknya gunakan pendekatan ini untuk mengakses layanan AWS, layanan pihak ketiga, dan layanan milik Anda yang di-host di VPC lain secara aman. Semua lalu lintas jaringan di AWS PrivateLink tetap dalam jalur utama AWS global dan tidak akan melintasi internet. Konektivitas hanya dapat diinisiasi oleh konsumen layanan, bukan oleh penyedia layanan. Dengan AWS PrivateLink untuk akses layanan eksternal, Anda dapat menciptakan VPC terisolasi tanpa akses internet dan membantu melindungi VPC Anda dari vektor ancaman eksternal. Layanan pihak ketiga dapat menggunakan AWS PrivateLink agar konsumen dapat terhubung ke layanan dari VPC mereka melalui alamat IP privat. Untuk aset VPC yang perlu membuat koneksi keluar ke internet, ini dapat ditetapkan khusus keluar (satu jalur) melalui gateway NAT yang dikelola AWS, gateway internet khusus keluar, atau proksi web yang Anda buat dan kelola.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Tinggi

Panduan implementasi

- Kontrol lalu lintas jaringan di VPC: Implementasikan praktik terbaik VPC untuk mengontrol lalu lintas.
  - Keamanan Amazon VPC
  - titik akhir VPC
  - Grup keamanan Amazon VPC
  - ACL jaringan
- Kontrol lalu lintas di edge: Implementasikan layanan edge, seperti Amazon CloudFront, untuk memberikan lapisan perlindungan tambahan dan fitur lainnya.
  - Kasus penggunaan Amazon CloudFront
  - AWS Global Accelerator
  - Firewall Aplikasi Web AWS (AWS WAF)
  - Amazon Route 53
  - Perutean Masuk Amazon VPC
- Kontrol lalu lintas jaringan privat: Implementasikan layanan yang melindungi lalu lintas privat untuk beban kerja Anda.
  - Peering Amazon VPC
  - Layanan Titik Akhir Amazon VPC (AWS PrivateLink)
  - Amazon VPC Transit Gateway

- AWS Direct Connect
- AWS Site-to-Site VPN
- AWS Client VPN
- Amazon S3 Access Points

Sumber daya

Dokumen terkait:

- AWS Firewall Manager
- Amazon Inspector
- Mulai menggunakan AWS WAF

Video terkait:

- Arsitektur referensi AWS Transit Gateway untuk banyak VPC
- Perlindungan dan Akselerasi Aplikasi dengan Amazon CloudFront, AWS WAF, dan AWS Shield

Contoh terkait:

- Lab: Deployment Otomatis VPC

SEC05-BP03 Mengotomatiskan perlindungan jaringan

Otomatiskan mekanisme perlindungan untuk memberikan jaringan perlindungan mandiri berdasarkan deteksi anomali dan kecerdasan ancaman. Misalnya, deteksi gangguan dan alat pencegahan yang dapat beradaptasi dengan ancaman masa kini serta mengurangi dampaknya. Firewall aplikasi web adalah contoh tempat Anda mengotomatiskan perlindungan jaringan, misalnya, dengan menggunakan solusi Otomatisasi Keamanan AWS WAF (https://github.com/awslabs/aws-waf-security-automations) untuk secara otomatis memblokir permintaan yang berasal dari alamat IP yang terkait dengan penyebab ancaman yang diketahui.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

- Otomatiskan perlindungan untuk lalu lintas berbasis web: AWS menawarkan solusi yang menggunakan AWS CloudFormation untuk secara otomatis melakukan deployment rangkaian aturan AWS WAF yang didesain untuk memfilter serangan berbasis web yang umum. Pengguna dapat memilih dari fitur perlindungan sebelum dikonfigurasi yang menentukan aturan yang disertakan dalam daftar kontrol akses web (web ACL) AWS WAF.
  - Otomatisasi keamanan AWS WAF
- Pertimbangkan solusi AWS Partner: Partner AWS menawarkan ratusan produk industri terkemuka yang setara, serupa, atau dapat diintegrasikan dengan kontrol yang sudah ada di lingkungan on-premise Anda. Produk-produk ini melengkapi layanan AWS untuk memungkinkan Anda melakukan deployment arsitektur keamanan yang menyeluruh dan pengalaman yang lancar di seluruh lingkungan cloud dan on-premise Anda.
  - Keamanan infrastruktur

Sumber daya

Dokumen terkait:

- AWS Firewall Manager
- Amazon Inspector
- Keamanan Amazon VPC
- Mulai menggunakan AWS WAF

Video terkait:

- AWS Transit Gateway reference architectures for many VPCs
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield

Contoh terkait:

- Lab: Deployment Otomatis VPC

SEC05-BP04 Mengimplementasikan inspeksi dan perlindungan

Inspeksi dan filter lalu lintas Anda di setiap lapisan. Anda dapat melakukan inspeksi terhadap konfigurasi VPC untuk potensi akses yang tidak diinginkan menggunakan VPC Network Access Analyzer. Anda dapat menentukan persyaratan akses jaringan Anda serta mengidentifikasi jalur jaringan yang berpotensi tidak memenuhi syarat tersebut. Untuk komponen transaksi melalui protokol berbasis HTTP, firewall aplikasi web dapat membantu melindungi dari serangan yang umum. AWS WAF adalah firewall aplikasi web yang memungkinkan Anda untuk memantau dan memblokir permintaan HTTP sesuai dengan aturan yang dapat Anda konfigurasikan yang diteruskan ke API Amazon API Gateway, Amazon CloudFront, atau Application Load Balancer. Untuk mulai menggunakan AWS WAF, Anda dapat memulai dengan Peraturan yang Dikelola AWS yang digabungkan dengan milik Anda sendiri, atau gunakan integrasi partner yang ada..

Untuk mengelola AWS WAF, perlindungan AWS Shield Advanced, dan grup keamanan Amazon VPC di seluruh AWS Organizations, Anda dapat menggunakan AWS Firewall Manager. Hal ini memungkinkan Anda untuk mengonfigurasikan dan mengelola aturan firewall di seluruh akun dan aplikasi Anda secara terpusat, sehingga lebih mudah untuk menskalakan penerapan aturan umum. Penerapan ini juga memungkinkan Anda merespons serangan dengan cepat, menggunakan AWS Shield Advanced, atau solusi yang dapat secara otomatis memblokir permintaan yang tidak diinginkan ke aplikasi web Anda. Firewall Manager juga menerapkan AWS Network Firewall. AWS Network Firewall adalah layanan terkelola yang menggunakan mesin aturan untuk memberi Anda kontrol fine-grained terhadap lalu lintas jaringan stateful dan stateless. Layanan ini mendukung spesifikasi sistem perlindungan gangguan (IPS) sumber terbuka yang sesuai dengan aturan Suricata untuk aturan yang membantu melindungi beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Konfigurasikan Amazon GuardDuty: GuardDuty adalah layanan pendeteksi ancaman yang secara terus-menerus memantau aktivitas berbahaya dan perilaku yang tidak diotorisasi untuk melindungi Akun AWS dan beban kerja Anda. Aktifkan GuardDuty dan konfigurasikan peringatan yang diotomatiskan.
  - Amazon GuardDuty
  - Lab: Deployment Otomatis Kontrol Deteksi
- Konfigurasikan Log Alur cloud privat virtual (VPC): Log Alur VPC adalah fitur yang memungkinkan Anda untuk mendokumentasikan informasi tentang lalu lintas IP ke dan dari antarmuka jaringan

di VPC Anda. Data log alur dapat dipublikasikan ke Amazon CloudWatch Logs dan Amazon Simple Storage Service (Amazon S3). Setelah Anda membuat log alur, Anda dapat mengambil dan melihat datanya di lokasi tujuan yang telah dipilih.

- Pertimbangkan traffic mirroring VPC: Traffic mirroring adalah fitur Amazon VPC yang dapat Anda gunakan untuk menyalin lalu lintas jaringan dari antarmuka jaringan elastis instans Amazon Elastic Compute Cloud (Amazon EC2) dan mengirimnya ke alat pemantauan dan keamanan luar jaringan untuk inspeksi konten, pemantauan ancaman, dan pemecahan masalah.
  - Traffic mirroring VPC

Sumber daya

Dokumen terkait:

- AWS Firewall Manager
- Amazon Inspector
- Keamanan Amazon VPC
- Mulai menggunakan AWS WAF

Video terkait:

- Arsitektur referensi AWS Transit Gateway untuk banyak VPC
- Perlindungan dan Akselerasi Aplikasi dengan Amazon CloudFront, AWS WAF, dan AWS Shield

Contoh terkait:

- Lab: Deployment Otomatis VPC

Melindungi komputasi

Sumber daya komputasi meliputi instans EC2, kontainer, fungsi AWS Lambda, layanan basis data, perangkat IoT, dan banyak lagi. Tiap-tiap tipe sumber daya komputasi ini memerlukan pendekatan yang berbeda-beda untuk mengamankannya. Namun, semuanya memiliki strategi yang sama yang perlu Anda pertimbangkan: pertahanan mendalam, manajemen kerentanan, pengurangan permukaan serangan, otomatisasi konfigurasi dan operasi, dan melakukan tindakan dari jarak jauh. Pada bagian ini, Anda akan menemukan panduan umum dalam melindungi sumber daya komputasi

untuk layanan-layanan utama Anda. Untuk tiap-tiap layanan AWS yang digunakan, penting bagi Anda untuk memeriksa saran keamanan khusus di dalam dokumentasi layanan.

Praktik terbaik

- SEC06-BP01 Melakukan manajemen kerentanan
- SEC06-BP02 Mengurangi permukaan serangan
- SEC06-BP03 Mengimplementasikan layanan terkelola
- SEC06-BP04 Mengotomatiskan perlindungan komputasi
- SEC06-BP05 Memberikan kemampuan melakukan tindakan dari jarak jauh
- SEC06-BP06 Memvalidasi integritas perangkat lunak

SEC06-BP01 Melakukan manajemen kerentanan

Seringlah memindai dan melakukan patching kerentanan pada kode, dependensi, dan infrastruktur Anda untuk membantu mencegah ancaman baru.

Hasil yang diinginkan: Membuat dan memelihara program manajemen kerentanan: Memindai dan melakukan patch sumber daya secara rutin, seperti instans Amazon EC2, kontainer Amazon Elastic Container Service (Amazon ECS), dan beban kerja Amazon Elastic Kubernetes Service (Amazon EKS). Mengonfigurasi jendela pemeliharaan untuk sumber daya yang dikelola AWS, seperti basis data Amazon Relational Database Service (Amazon RDS). Menggunakan pemindaian kode statis untuk memeriksa kode sumber aplikasi untuk masalah umum. Pertimbangkan uji penetrasi aplikasi web jika organisasi Anda memiliki keterampilan yang diperlukan atau dapat menggunakan bantuan dari luar.

Antipola umum:

- Tidak memiliki program manajemen kerentanan.
- Menjalankan patching sistem tanpa mempertimbangkan tingkat keparahan atau penghindaran risiko.
- Menggunakan perangkat lunak yang sudah lewat tanggal akhir masa pakai (EOL) dari vendor.
- Melakukan deployment kode ke dalam produksi sebelum menganalisis masalah keamanan.

Manfaat menjalankan praktik terbaik ini:

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Program manajemen kerentanan mencakup penilaian keamanan, mengidentifikasi masalah, memprioritaskan, dan menjalankan operasi patch sebagai bagian dari penyelesaian masalah. Otomatisasi adalah kunci agar dapat terus memindai beban kerja untuk menemukan masalah dan paparan jaringan yang tidak diinginkan serta melakukan perbaikan. Mengotomatiskan pembuatan dan pembaruan sumber daya menghemat waktu dan mengurangi risiko kesalahan konfigurasi yang menciptakan masalah lebih lanjut. Program manajemen kerentanan yang dirancang dengan baik juga harus mempertimbangkan pengujian kerentanan selama tahap pengembangan dan deployment siklus hidup perangkat lunak. Mengimplementasikan manajemen kerentanan selama pengembangan dan deployment membantu mengurangi kemungkinan kerentanan dapat masuk ke lingkungan produksi Anda.

Mengimplementasikan program manajemen kerentanan memerlukan pemahaman yang baik terkait Model Tanggung Jawab Bersama AWS dan bagaimana kaitannya dengan beban kerja tertentu. Dalam Model Tanggung Jawab Bersama, AWS bertanggung jawab untuk melindungi infrastruktur AWS Cloud. Infrastruktur ini terdiri dari perangkat keras, perangkat lunak, jaringan, dan fasilitas yang menjalankan layanan AWS Cloud. Anda bertanggung jawab atas keamanan di cloud, misalnya, data aktual, konfigurasi keamanan, dan tugas manajemen instans Amazon EC2, serta memastikan bahwa klasifikasi dan konfigurasi objek Amazon S3 Anda sudah tepat. Pendekatan terhadap manajemen kerentanan juga dapat bervariasi, bergantung pada layanan yang digunakan. Misalnya, AWS mengelola patching untuk layanan basis data relasional terkelola kami, Amazon RDS, tetapi Anda akan bertanggung jawab atas patching basis data yang di-hosting secara mandiri.

AWS memiliki berbagai layanan untuk membantu program manajemen kerentanan. Amazon Inspector terus memindai beban kerja AWS untuk menemukan masalah perangkat lunak dan akses jaringan yang tidak diinginkan. AWS Systems Manager Patch Manager membantu mengelola patching di seluruh instans Amazon EC2. Amazon Inspector dan Systems Manager dapat dilihat di AWS Security Hub, layanan manajemen postur keamanan cloud yang membantu mengotomatiskan pemeriksaan keamanan AWS dan memusatkan peringatan keamanan.

Amazon CodeGuru dapat membantu mengidentifikasi potensi masalah di aplikasi Java dan Python menggunakan analisis kode statis.

Langkah implementasi

- Konfigurasikan Amazon Inspector: Amazon Inspector secara otomatis mendeteksi instans Amazon EC2 yang baru saja diluncurkan, fungsi Lambda, dan gambar kontainer yang dimasukkan ke

Amazon ECR serta segera memindainya untuk menemukan masalah perangkat lunak, potensi kecacatan, dan paparan jaringan yang tidak diinginkan.

- Pindai kode sumber: Pindai pustaka dan dependensi untuk menemukan masalah dan kecacatan. Amazon CodeGuru dapat memindai dan memberikan rekomendasi untuk memperbaiki masalah keamanan umum untuk aplikasi Java dan Python. OWASP Foundation menerbitkan daftar Alat Analisis Kode Sumber (juga disebut sebagai alat SAST).
- Implementasikan mekanisme untuk memindai dan melakukan patching lingkungan yang ada, serta pemindaian sebagai bagian dari proses pembuatan jalur CI/CD: Implementasikan mekanisme untuk memindai dan melakukan patching masalah di dependensi dan sistem operasi untuk membantu melindungi dari ancaman baru. Jalankan mekanisme tersebut secara rutin. Penting untuk memahami manajemen kerentanan perangkat lunak saat Anda ingin menerapkan patch atau mengatasi masalah perangkat lunak. Prioritaskan perbaikan potensi masalah keamanan dengan menanamkan penilaian kerentanan lebih awal ke dalam jalur integrasi berkelanjutan/pengiriman berkelanjutan (CI/CD). Pendekatan dapat bervariasi berdasarkan layanan AWS yang digunakan. Untuk memeriksa potensi masalah terkait perangkat lunak yang dijalankan di instans Amazon EC2, tambahkan Amazon Inspector ke jalur untuk memberikan peringatan dan menghentikan proses pembuatan jika masalah atau potensi kecacatan terdeteksi. Amazon Inspector memantau sumber daya secara berkelanjutan. Anda juga dapat menggunakan produk sumber terbuka seperti Pemeriksaan Dependensi OWASP, Snyk, OpenVAS, manajer paket, dan alat AWS Partner untuk manajemen kerentanan.
- Gunakan AWS Systems Manager: Anda bertanggung jawab atas manajemen patch sumber daya AWS Anda, termasuk instans Amazon Elastic Compute Cloud (Amazon EC2), Amazon Machine Image (AMI), dan sumber daya komputasi lainnya. AWS Systems Manager Patch Manager mengotomatiskan proses patching instans terkelola dengan pembaruan terkait keamanan dan jenis pembaruan lainnya. Patch Manager dapat digunakan untuk menerapkan patch pada instans Amazon EC2 untuk sistem operasi dan aplikasi, termasuk aplikasi Microsoft, paket layanan Windows, dan pembaruan versi minor untuk instans berbasis Linux. Selain Amazon EC2, Patch Manager juga dapat digunakan untuk melakukan patching server on-premise.

  Untuk daftar sistem operasi yang didukung, lihat Sistem operasi yang didukung di Panduan Pengguna Systems Manager. Anda dapat memindai instans untuk hanya melihat laporan patch yang hilang, atau Anda dapat memindai dan secara otomatis memasang semua patch yang hilang.
- Gunakan AWS Security Hub: Security Hub menyediakan tampilan komprehensif untuk status keamanan Anda di AWS. Program ini mengumpulkan data keamanan di berbagai layanan AWS dan mengumpulkan temuan tersebut dalam format standar, memungkinkan Anda untuk memprioritaskan temuan keamanan di seluruh layanan AWS.

- Gunakan AWS CloudFormation: AWS CloudFormation adalah layanan infrastruktur sebagai kode (IaC) yang dapat membantu manajemen kerentanan dengan mengotomatiskan deployment sumber daya dan menstandarkan arsitektur sumber daya di berbagai akun dan lingkungan.

Sumber daya

Dokumen terkait:

- AWS Systems Manager
- Gambaran Umum Keamanan AWS Lambda
- Amazon CodeGuru
- Manajemen Kerentanan Otomatis yang Ditingkatkan untuk Beban Kerja Cloud dengan Amazon Inspector Baru
- Mengotomatiskan manajemen kerentanan dan perbaikan di AWS menggunakan Amazon Inspector dan AWS Systems Manager – Bagian 1

Video terkait:

- Mengamankan Layanan Kontainer dan Nirserver
- Praktik terbaik keamanan untuk layanan metadata instans Amazon EC2

SEC06-BP02 Mengurangi permukaan serangan

Kurangi paparan Anda ke akses yang tidak diinginkan dengan penguatan sistem operasi serta meminimalkan penggunaan komponen, pustaka, dan layanan sekali pakai eksternal. Mulai dengan mengurangi komponen yang tidak digunakan untuk seluruh beban kerja, baik itu paket sistem operasi atau aplikasi, untuk beban kerja berbasis Amazon Elastic Compute Cloud (Amazon EC2), maupun modul perangkat lunak eksternal pada kode Anda. Anda dapat menemukan beberapa panduan konfigurasi penguatan dan keamanan untuk sistem operasi umum dan perangkat lunak server. Misalnya, Anda dapat memulai dengan Pusat Keamanan Internet dan lakukan iterasi.

Di Amazon EC2, Anda dapat membuat Amazon Machine Image (AMI) Anda sendiri, yang telah di-patch dan diperkuat, untuk membantu memenuhi persyaratan keamanan spesifik bagi organisasi Anda. Patch dan kontrol keamanan lain yang Anda terapkan di AMI efektif pada saat dibuat—sifatnya tidak dinamis kecuali Anda memodifikasinya setelah peluncuran, misalnya, dengan AWS Systems Manager.

Anda dapat menyederhanakan proses membangun AMI yang aman dengan EC2 Image Builder. EC2 Image Builder secara signifikan mengurangi upaya yang diperlukan untuk membuat dan memelihara image emas tanpa otomatisasi penulisan dan pemeliharaan. Ketika pembaruan perangkat lunak sudah tersedia, Image Builder secara otomatis memproduksi image baru tanpa mengharuskan pengguna memulai pembangunan image secara manual. EC2 Image Builder memungkinkan Anda untuk memvalidasi fungsionalitas dan keamanan image Anda dengan mudah sebelum menggunakannya dalam produksi dengan pengujian yang disediakan AWS serta pengujian Anda sendiri. Anda juga dapat menerapkan pengaturan keamanan yang disediakan AWS untuk mengamankan image Anda secara lebih lanjut untuk memenuhi kriteria keamanan internal. Misalnya, Anda dapat memproduksi image yang sesuai dengan standar Security Technical Implementation Guide (STIG) menggunakan templat yang disediakan oleh AWS.

Menggunakan alat analisis kode statis pihak ketiga, Anda dapat mengidentifikasi masalah keamanan umum seperti batas input fungsi yang tidak diperiksa, dan juga kelemahan dan paparan umum (common vulnerabilities and exposures, CVE) yang dapat diterapkan. Anda dapat menggunakan Amazon CodeGuru untuk bahasa yang didukung. Alat pemeriksaan dependensi juga dapat digunakan untuk menentukan apakah pustaka yang ditautkan kode Anda merupakan versi terbaru, bebas dari CVE, serta memiliki syarat perizinan yang memenuhi persyaratan kebijakan perangkat lunak Anda.

Menggunakan Amazon Inspector, Anda dapat melakukan penilaian konfigurasi terhadap instans Anda untuk CVE yang diketahui, menilai tolok ukur keamanan, serta mengotomatiskan pemberitahuan kecacatan. Amazon Inspector berjalan di instans produksi atau di jalur build, serta memberi tahu pengembang dan teknisi ketika ada temuan. Anda dapat mengakses temuan secara terprogram dan mengarahkan tim Anda ke backlog dan sistem pelacakan bug. EC2 Image Builder dapat digunakan untuk memelihara image server (AMI) dengan patching otomatis, penegakan kebijakan keamanan yang disediakan oleh AWS, serta kustomisasi lainnya. Ketika menggunakan kontainer, terapkan Pemindaian Image ECR pada jalur build Anda dan secara rutin terhadap repositori image Anda untuk mencari CVE dalam kontainer Anda.

Ketika Amazon Inspector dan alat lainnya mengidentifikasi dengan efektif konfigurasi dan CVE apa pun yang ada, metode lain diperlukan untuk menguji beban kerja Anda pada tingkat aplikasi. Fuzzing adalah metode yang terkenal untuk menemukan bug menggunakan otomatisasi untuk menginjeksi data dengan kesalahan bentuk ke bidang input dan area lain pada aplikasi Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

- Perkuat sistem operasi: Konfigurasikan sistem operasi agar memenuhi praktik terbaik.
  - Mengamankan Amazon Linux
  - Mengamankan Microsoft Windows Server
- Perkuat sumber daya terkontainerisasi: Konfigurasikan sumber daya terkontainerisasi untuk memenuhi praktik terbaik keamanan.
- Terapkan praktik terbaik AWS Lambda.
  - Praktik terbaik AWS Lambda

Sumber daya

Dokumen terkait:

- AWS Systems Manager
- Mengganti Host Bastion dengan Amazon EC2 Systems Manager
- Gambaran Umum Keamanan AWS Lambda

Video terkait:

- Menjalankan beban kerja dengan keamanan tinggi di Amazon EKS
- Mengamankan Layanan Kontainer dan Nirserver
- Praktik terbaik keamanan untuk layanan metadata instans Amazon EC2

Contoh terkait:

- Lab: Deployment Firewall Aplikasi Web secara Otomatis

SEC06-BP03 Mengimplementasikan layanan terkelola

Implementasikan layanan yang mengelola sumber daya seperti Amazon Relational Database Service (Amazon RDS), AWS Lambda, dan Amazon Elastic Container Service (Amazon ECS), untuk mengurangi tugas pemeliharaan keamanan sebagai bagian dari model tanggung jawab bersama. Contohnya, Amazon RDS membantu Anda mengatur, mengoperasikan, dan menskalakan basis data relasional, mengotomatiskan tugas administrasi seperti penyediaan perangkat keras, pengaturan

basis data, patching, dan pencadangan. Ini berarti Anda memiliki lebih banyak waktu luang untuk berkonsentrasi mengamankan aplikasi Anda dengan cara lain yang disebutkan dalam AWS Well-Architected Framework. Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server, sehingga Anda hanya perlu fokus pada konektivitas, permintaan, dan keamanan di tingkat kode–bukan infrastruktur atau sistem operasi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

- Jelajahi layanan yang tersedia: Jelajahi, uji, dan terapkan layanan yang mengelola sumber daya, seperti Amazon RDS, AWS Lambda, dan Amazon ECS.

Sumber daya

Dokumen terkait:

- Situs web AWS
- AWS Systems Manager
- Mengganti Host Bastion dengan Amazon EC2 Systems Manager
- Gambaran Umum Keamanan AWS Lambda

Video terkait:

- Menjalankan beban kerja dengan keamanan tinggi di Amazon EKS
- Mengamankan Layanan Kontainer dan Nirserver
- Praktik terbaik keamanan untuk layanan metadata instans Amazon EC2

Contoh terkait:

- Lab: Permohonan Sertifikat Publik AWS Certificate Manager

SEC06-BP04 Mengotomatiskan perlindungan komputasi

Otomatiskan mekanisme komputasi protektif Anda termasuk manajemen kelemahan, pengurangan permukaan serangan, serta manajemen sumber daya. Otomatisasi akan membantu Anda

menginvestasikan waktu untuk mengamankan aspek-aspek lain dalam beban kerja Anda, dan mengurangi risiko kesalahan manusia.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

- Otomatiskan manajemen konfigurasi: Tegakkan dan validasi konfigurasi keamanan secara otomatis menggunakan layanan atau alat manajemen konfigurasi.
  - AWS Systems Manager
  - AWS CloudFormation
  - Lab: Deployment otomatis VPC
  - Lab: Deployment otomatis aplikasi web EC2

- Otomatiskan patching instans Amazon Elastic Compute Cloud (Amazon EC2): AWS Systems Manager Patch Manager mengotomatiskan proses patching instans terkelola dengan pembaruan terkait keamanan dan jenis pembaruan lainnya. Anda dapat menggunakan Patch Manager guna menerapkan patch untuk sistem operasi maupun aplikasi.
  - AWS Systems Manager Patch Manager
  - Patching multiakun dan multiwilayah terpusat dengan AWS Systems Manager Automation

- Implementasikan deteksi dan pencegahan intrusi: Implementasikan alat deteksi dan pencegahan intrusi untuk memantau dan menghentikan aktivitas berbahaya pada instans.
- Pertimbangkan solusi AWS Partner: Partner AWS menawarkan ratusan produk industri terkemuka yang setara, serupa, atau dapat diintegrasikan dengan kontrol yang sudah ada di lingkungan on-premise Anda. Produk-produk ini melengkapi layanan AWS untuk memungkinkan Anda melakukan deployment arsitektur keamanan yang menyeluruh dan pengalaman yang lancar di seluruh lingkungan cloud dan on-premise Anda.
  - Keamanan infrastruktur

Sumber daya

Dokumen terkait:

- AWS CloudFormation

- AWS Systems Manager
- AWS Systems Manager Patch Manager
- Patching multiakun dan multiwilayah terpusat dengan AWS Systems Manager Automation
- Keamanan infrastruktur
- Mengganti Host Bastion dengan Amazon EC2 Systems Manager
- Gambaran Umum Keamanan AWS Lambda

Video terkait:

- Running high-security workloads on Amazon EKS
- Securing Serverless and Container Services
- Security best practices for the Amazon EC2 instance metadata service

Contoh terkait:

- Lab: Deployment Firewall Aplikasi Web secara Otomatis
- Lab: Deployment otomatis aplikasi web EC2

SEC06-BP05 Memberikan kemampuan melakukan tindakan dari jarak jauh

Menghapus kemampuan akses interaktif dapat mengurangi risiko kesalahan akibat kelalaian manusia, dan kemungkinan dibutuhkannya manajemen atau konfigurasi manual. Misalnya, gunakan alur kerja manajemen perubahan untuk melakukan deployment instans Amazon Elastic Compute Cloud (Amazon EC2) menggunakan infrastruktur sebagai kode, selanjutnya kelola instans Amazon EC2 menggunakan alat seperti AWS Systems Manager, bukannya menerapkan akses langsung melalui host bastion. AWS Systems Managerdapat mengotomatiskan berbagai tugas pemeliharaan dan deployment, menggunakan fitur yang mencakup alur kerja otomatisasi, dokumen (buku pedoman), dan run command (jalankan perintah). Tumpukan AWS CloudFormation dibangun dari pipeline dan dapat mengotomatiskan tugas manajemen serta deployment infrastruktur Anda tanpa menggunakan AWS Management Console atau API secara langsung.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Ganti akses konsol: Ganti akses konsol (SSH atau RDP) ke instans dengan AWS Systems Manager Run Command untuk mengotomatiskan tugas manajemen.

- AWS Systems Manager Run Command

Sumber daya

Dokumen terkait:

- AWS Systems Manager
- AWS Systems Manager Run Command
- Mengganti Host Bastion dengan Amazon EC2 Systems Manager
- Gambaran Umum Keamanan AWS Lambda

Video terkait:

- Jalankan beban kerja dengan keamanan tinggi di Amazon EKS
- Mengamankan Layanan Kontainer dan Nirserver
- Praktik terbaik keamanan untuk layanan metadata instans Amazon EC2

Contoh terkait:

- Lab: Deployment Otomatis Firewall Aplikasi Web

SEC06-BP06 Memvalidasi integritas perangkat lunak

Implementasikan mekanisme (misalnya, penandatanganan kode) untuk memvalidasi bahwa perangkat lunak, kode, dan pustaka yang digunakan di beban kerja berasal dari sumber tepercaya dan belum pernah dimodifikasi. Misalnya, Anda harus memverifikasi sertifikat penandatanganan kode biner dan skrip untuk mengonfirmasi penulis, serta memastikan sertifikat tersebut belum pernah dimodifikasi sejak dibuat oleh penulisnya. AWS Signer dapat membantu memastikan kepercayaan dan integritas kode Anda dengan mengelola secara terpusat siklus hidup penandatanganan kode, termasuk sertifikat penandatanganan serta kunci privat dan publik. Anda dapat mempelajari cara

menggunakan pola tingkat lanjut dan praktik terbaik penandatanganan kode dengan AWS Lambda. Selain itu, checksum perangkat lunak yang Anda unduh, dibandingkan dengan checksum dari penyedia, dapat membantu memastikan bahwa perangkat belum pernah dimodifikasi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Selidiki mekanisme: Penandatanganan kode adalah sebuah mekanisme yang dapat digunakan untuk memvalidasi integritas perangkat lunak.
  - NIST: Pertimbangan Keamanan untuk Penandatanganan Kode

Sumber daya

Dokumen terkait:

- AWS Signer
- Baru – Penandatanganan Kode, Kontrol Integritas dan Kepercayaan untuk AWS Lambda

Perlindungan data

Sebelum merancang beban kerja apa pun, praktik mendasar yang berpengaruh terhadap keamanan harus diterapkan. Misalnya, klasifikasi data menjadi cara untuk mengategorikan data berdasarkan tingkat sensitivitas, dan enkripsi melindungi data dengan membuatnya tidak dapat dikenali oleh akses tidak sah. Metode ini penting karena dapat mendukung tujuan seperti mencegah kesalahan penanganan atau mematuhi kewajiban peraturan.

Di AWS, ada berbagai pendekatan yang dapat Anda gunakan saat menangani perlindungan data. Bagian berikut menjelaskan cara menggunakan pendekatan ini:

Topik

Klasifikasi data

Klasifikasi data menyediakan cara untuk mengategorikan data organisasi berdasarkan kekritisan dan sensitivitas untuk membantu Anda menentukan kontrol retensi dan perlindungan yang sesuai.

Praktik terbaik

SEC07-BP01 Mengidentifikasi data dalam beban kerja Anda

Penting untuk memahami jenis dan klasifikasi data yang sedang diproses oleh beban kerja Anda, proses bisnis terkait, tempat penyimpanan data, dan pemilik data. Anda juga harus memahami persyaratan hukum dan kepatuhan yang berlaku dari beban kerja Anda, dan kontrol data apa yang perlu diterapkan. Mengidentifikasi data adalah langkah pertama dalam perjalanan klasifikasi data.

Manfaat menjalankan praktik terbaik ini:

Dengan klasifikasi data, pemilik beban kerja dapat mengidentifikasi lokasi penyimpanan data sensitif dan menentukan bagaimana data tersebut dapat diakses dan dibagikan.

Klasifikasi data bertujuan untuk menjawab pertanyaan berikut:

- Jenis data apa yang Anda miliki?

  Data dapat berupa:

  - Kekayaan intelektual (IP), seperti rahasia dagang, paten, atau perjanjian kontrak.
  - Informasi kesehatan yang dilindungi (PHI), seperti rekam medis yang berisi informasi riwayat kesehatan seseorang.
  - Informasi pengenal pribadi (PII), seperti nama, alamat, tanggal lahir, dan nomor registrasi atau ID nasional.
  - Data kartu kredit, seperti Nomor Rekening Utama (PAN), nama pemilik kartu, tanggal kedaluwarsa, dan nomor kode layanan.
  - Di mana data sensitif disimpan?
  - Siapa yang dapat mengakses, mengubah, dan menghapus data?
  - Memahami izin pengguna adalah hal yang penting dalam menghindari potensi kesalahan penanganan data.
- Siapa yang dapat melakukan operasi membuat, membaca, memperbarui, dan menghapus (CRUD)?
  - Antisipasi potensi peningkatan hak akses dengan memahami siapa yang dapat mengelola izin ke data.
- Dampak bisnis seperti apa yang mungkin terjadi jika data secara tidak sengaja diungkapkan, diubah, atau dihapus?
  - Pahami konsekuensi risiko jika data diubah, dihapus, atau diungkapkan secara tidak sengaja.

Dengan mengetahui jawaban atas pertanyaan ini, Anda dapat mengambil tindakan berikut:

- Mengurangi cakupan data sensitif (seperti jumlah lokasi data sensitif) dan membatasi akses ke data sensitif hanya untuk pengguna yang disetujui.
- Memahami berbagai jenis data sehingga Anda dapat menerapkan mekanisme dan teknik perlindungan data yang sesuai, seperti enkripsi, pencegahan kehilangan data, serta manajemen identitas dan akses.
- Mengoptimalkan biaya dengan memberikan tujuan kontrol yang tepat untuk data.

- Tanpa ragu menjawab pertanyaan dari regulator dan auditor mengenai jenis dan jumlah data, dan bagaimana data dengan sensitivitas yang berbeda dipisahkan dari satu sama lain.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Klasifikasi data adalah tindakan mengidentifikasi sensitivitas data. Aktivitas ini mungkin melibatkan pemberian tag untuk memudahkan pencarian dan pelacakan data. Klasifikasi data juga mengurangi duplikasi data, yang dapat membantu mengurangi biaya penyimpanan dan pencadangan sekaligus mempercepat proses pencarian.

Gunakan layanan seperti Amazon Macie untuk mengotomatiskan penemuan dan klasifikasi data sensitif dalam skala besar. Layanan lain, seperti Amazon EventBridge dan AWS Config, dapat digunakan untuk mengotomatiskan perbaikan masalah keamanan data, seperti bucket Amazon Simple Storage Service (Amazon S3) tidak terenkripsi dan volume EBS Amazon EC2 atau sumber daya data yang tidak diberi tag. Untuk daftar lengkap integrasi layanan AWS, lihat dokumentasi EventBridge.

Mendeteksi PII dalam data yang tidak terstruktur seperti email pelanggan, tiket dukungan, ulasan produk, dan media sosial dapat dilakukan menggunakan Amazon Comprehend, yang merupakan layanan pemrosesan bahasa alami (NLP) yang menggunakan machine learning (ML) untuk menemukan wawasan dan hubungan seperti orang, tempat, sentimen, serta topik dalam teks yang tidak terstruktur. Untuk daftar layanan AWS yang dapat membantu identifikasi data, lihat Teknik umum untuk mendeteksi data PHI dan PII menggunakan layanan AWS.

Metode lain yang mendukung klasifikasi dan perlindungan data adalah Pemberian tag sumber daya AWS. Pemberian tag memungkinkan Anda menetapkan metadata ke sumber daya AWS yang dapat digunakan untuk mengelola, mengidentifikasi, mengatur, mencari, dan memfilter sumber daya.

Dalam beberapa kasus, Anda mungkin memilih untuk memberi tag seluruh sumber daya (seperti bucket S3), khususnya saat beban kerja atau layanan tertentu diharapkan menyimpan proses atau transmisi dari klasifikasi data yang sudah umum.

Jika perlu, Anda dapat memberi tag bucket S3 dan bukan pada objek individu untuk kemudahan administrasi dan pemeliharaan keamanan.

Langkah implementasi

Deteksi data sensitif dalam Amazon S3:

1. Sebelum memulai, pastikan Anda memiliki izin yang sesuai untuk mengakses konsol Amazon Macie dan operasi API. Untuk detail tambahan, lihat Mulai menggunakan Amazon Macie.
2. Gunakan Amazon Macie untuk menjalankan penemuan data otomatis saat data sensitif berada di Amazon S3.
   - Gunakan panduan Mulai Menggunakan Amazon Macie untuk mengonfigurasi repositori untuk hasil penemuan data sensitif dan membuat tugas penemuan untuk data sensitif.
   - Cara menggunakan Amazon Macie untuk pratinjau data sensitif di bucket S3.

     Secara default, Macie menganalisis objek menggunakan set pengidentifikasi data terkelola yang kami rekomendasikan untuk penemuan data sensitif otomatis. Anda dapat menyesuaikan analisis dengan mengonfigurasi Macie untuk menggunakan pengidentifikasi data terkelola tertentu, pengidentifikasi data kustom, dan daftar yang diizinkan saat melakukan penemuan data sensitif otomatis untuk akun atau organisasi Anda. Anda dapat menyesuaikan cakupan analisis dengan mengecualikan bucket tertentu (misalnya, bucket S3 yang biasanya menyimpan data pencatatan log AWS).
3. Untuk mengonfigurasi dan menggunakan penemuan data sensitif otomatis, lihat Menjalankan penemuan data sensitif otomatis dengan Amazon Macie.
4. Anda juga dapat mempertimbangkan Penemuan Data Otomatis untuk Amazon Macie.

Deteksi data sensitif dalam Amazon RDS:

Untuk informasi lebih lanjut tentang penemuan data di basis data Amazon Relational Database Service (Amazon RDS), lihat Mengaktifkan klasifikasi data untuk basis data Amazon RDS dengan Macie.

Deteksi data sensitif dalam DynamoDB:

- Mendeteksi data sensitif di DynamoDB dengan Macie menjelaskan cara menggunakan Amazon Macie untuk mendeteksi data sensitif di tabel Amazon DynamoDB dengan mengekspor data ke Amazon S3 untuk pemindaian.

Solusi Partner AWS:

- Pertimbangkan untuk menggunakan ekstensi AWS Partner Network. Partner AWS memiliki alat ekstensi dan kerangka kerja kepatuhan yang terintegrasi langsung dengan layanan AWS. Partner dapat memberikan solusi tata kelola dan kepatuhan yang disesuaikan untuk membantu memenuhi kebutuhan organisasi Anda.

- Untuk solusi kustom dalam klasifikasi data, lihat Tata kelola data dalam peraturan dan persyaratan kepatuhan.

Anda dapat secara otomatis menerapkan standar pemberian tag yang diadopsi oleh organisasi Anda dengan membuat dan melakukan deployment kebijakan menggunakan AWS Organizations. Kebijakan tag memungkinkan Anda menentukan aturan yang menentukan nama kunci yang valid dan nilai apa yang valid untuk setiap kunci. Anda dapat memilih untuk hanya memantau, yang dapat Anda gunakan untuk mengevaluasi dan menghapus tag yang ada. Setelah tag mematuhi standar yang dipilih, Anda dapat mengaktifkan penerapan di kebijakan tag untuk mencegah pembuatan tag yang tidak patuh. Untuk detail lebih lanjut, lihat Mengamankan tag sumber daya yang digunakan untuk otorisasi menggunakan kebijakan kontrol layanan di AWS Organizations dan contoh kebijakan di mencegah perubahan tag selain oleh pengguna utama yang sah.

- Untuk mulai menggunakan kebijakan tag di AWS Organizations, sangat disarankan untuk mengikuti alur kerja di Mulai menggunakan kebijakan tag sebelum melanjutkan ke kebijakan tag yang lebih tinggi. Memahami efek pelampiran kebijakan tag sederhana ke satu akun sebelum menerapkannya ke seluruh unit organisasi (OU) atau organisasi dapat memberikan gambaran akan efek kebijakan tag sebelum Anda menerapkan kepatuhan ke kebijakan tag. Mulai menggunakan kebijakan tag menyediakan tautan ke instruksi untuk tugas terkait kebijakan yang lebih tinggi.
- Pertimbangkan untuk mengevaluasi layanan dan fitur AWS lainnya yang mendukung klasifikasi data, yang tercantum dalam laporan resmi Klasifikasi Data.

Sumber daya

Dokumen terkait:

- Mulai Menggunakan Amazon Macie
- Penemuan data otomatis dengan Amazon Macie
- Mulai menggunakan kebijakan tag
- Mendeteksi entitas PII

Blog terkait:

- Cara menggunakan Amazon Macie untuk pratinjau data sensitif di bucket S3.
- Menjalankan penemuan data sensitif dengan Amazon Macie.

- Teknik umum untuk mendeteksi data PHI dan PII menggunakan Layanan AWS
- Mendeteksi dan mengedit PII menggunakan Amazon Comprehend
- Menggunakan tag sumber daya yang digunakan untuk otorisasi menggunakan kebijakan kontrol layanan di AWS Organizations
- Melakukan klasifikasi data untuk basis data Amazon RDS dengan Macie
- Mendeteksi data sensitif di DynamoDB dengan Macie
-

Video terkait:

- Keamanan yang diarahkan peristiwa menggunakan Amazon Macie
- Amazon Macie untuk perlindungan dan tata kelola data
- Menyaring temuan data sensitif dengan daftar yang diizinkan

SEC07-BP02 Menentukan kontrol perlindungan data

Lindungi data sesuai dengan tingkat klasifikasinya. Contohnya, amankan data dengan klasifikasi publik menggunakan rekomendasi yang relevan sekaligus melindungi data sensitif dengan kontrol tambahan.

Dengan menggunakan tag sumber daya, pisahkan AWSakun berdasarkan sensitivitas (dan berpotensi juga untuk setiap peringatan, enklave, atau komunitas minat), kebijakan IAM, SCP AWS Organizations, AWS Key Management Service (AWS KMS), dan AWS CloudHSM, Anda dapat menentukan dan menerapkan kebijakan Anda untuk klasifikasi dan perlindungan data dengan enkripsi. Contohnya, jika Anda memiliki proyek dengan bucket S3 yang berisi data yang sangat penting atau instans Amazon Elastic Compute Cloud (Amazon EC2) yang memproses data rahasia, Anda dapat menandainya dengan tag `Project=ABC` . Hanya tim langsung Anda yang mengetahui arti kode proyek ini, dan ini menyediakan cara untuk menggunakan kontrol akses berbasis atribut. Anda dapat menentukan tingkatan akses ke kunci enkripsi AWS KMS melalui kebijakan dan pemberian kunci untuk memastikan hanya layanan yang sesuai yang memiliki akses ke konten sensitif melalui mekanisme yang aman. Jika Anda membuat keputusan otorisasi berdasarkan tag, Anda harus memastikan bahwa izin pada tag telah ditentukan dengan tepat menggunakan kebijakan tag di AWS Organizations.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

- Tentukan identifikasi dan skema klasifikasi data Anda: Identifikasi dan klasifikasi data Anda dilakukan untuk menilai potensi dampak dan tipe data yang Anda simpan dan siapa saja yang dapat mengaksesnya.
  - Dokumentasi AWS
- Temukan kontrol AWS yang tersedia: Untuk layanan AWS yang sedang atau akan Anda gunakan, temukan kontrol keamanannya. Banyak layanan memiliki bagian keamanan dalam dokumentasinya.
  - Dokumentasi AWS
- Kenali sumber daya kepatuhan AWS: Kenali sumber daya yang disediakan oleh AWS untuk membantu.
  - https://aws.amazon.com/compliance/

Sumber daya

Dokumen terkait:

- Dokumentasi AWS
- Laporan Resmi Klasifikasi Data
- Mulai menggunakan Amazon Macie
- Teks Hilang

Video terkait:

- Memperkenalkan Amazon Macie Baru

SEC07-BP03 Mengotomatisasi identifikasi dan klasifikasi

Otomatisasi identifikasi dan klasifikasi data dapat membantu Anda mengimplementasikan kontrol yang tepat. Menggunakan otomatisasi untuk hal ini, sebagai ganti akses langsung dari orang, dapat mengurangi risiko kesalahan dan eksposur manusia. Anda harus mengevaluasi menggunakan alat, seperti Amazon Macie, yang menggunakan machine learning untuk menemukan, mengelompokkan, dan melindungi data sensitif secara otomatis di AWS. Amazon Macie mengenali data sensitif, seperti

informasi pengenal pribadi (PII) atau kekayaan intelektual, dan membekali Anda dengan dasbor dan peringatan yang memberikan visibilitas tentang bagaimana data ini diakses atau dipindahkan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

- Gunakan Inventaris Amazon Simple Storage Service (Amazon S3): Inventaris Amazon S3 adalah salah satu alat yang dapat Anda gunakan untuk mengaudit serta melaporkan replikasi dan status enkripsi objek.
  - Inventaris Amazon S3
- Pertimbangkan Amazon Macie: Amazon Macie menggunakan machine learning untuk secara otomatis menemukan dan mengelompokkan data yang disimpan di Amazon S3.
  - Amazon Macie

Sumber daya

Dokumen terkait:

- Amazon Macie
- Inventaris Amazon S3
- Laporan Resmi Klasifikasi Data
- Mulai menggunakan Amazon Macie

Video terkait:

- Memperkenalkan Amazon Macie Baru

SEC07-BP04 Menentukan manajemen siklus hidup data

Strategi siklus hidup yang ditentukan harus berdasarkan tingkat sensitivitas serta persyaratan hukum dan organisasi. Aspek-aspek yang perlu diperhatikan mencakup durasi mempertahankan data, proses penghancuran data, manajemen akses data, transformasi data, dan berbagi data. Saat memilih metodologi klasifikasi data, seimbangkan ketergunaan dan akses. Anda juga harus mengakomodasi beberapa tingkat akses dan perbedaan untuk mengimplementasikan pendekatan yang aman tetapi masih dapat digunakan untuk masing-masing tingkat. Selalu gunakan pendekatan

pertahanan mendalam dan kurangi akses manusia ke data dan mekanisme untuk mentransformasi, menghapus, atau menyalin data. Misalnya, pengguna perlu diautentikasi oleh aplikasi, dan berikan izin akses yang diperlukan untuk melakukan tindakan dari jarak jauh kepada aplikasi, bukan pengguna. Selain itu, pastikan bahwa pengguna berasal dari jalur jaringan tepercaya dan memerlukan akses ke kunci dekripsi. Gunakan alat seperti dasbor dan pelaporan otomatis untuk memberikan informasi data kepada pengguna daripada memberi mereka akses langsung ke data.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Identifikasikan jenis data: Identifikasikan jenis data yang disimpan atau diproses di beban kerja. Data tersebut dapat berupa teks, gambar, basis data biner, dan lainnya.

Sumber daya

Dokumen terkait:

- Laporan Resmi Klasifikasi Data
- Mulai menggunakan Amazon Macie

Video terkait:

- Memperkenalkan Amazon Macie Baru

Lindungi data diam

Data diam merepresentasikan data apa pun yang Anda pertahankan di penyimpanan non-volatile selama durasi apa pun di beban kerja Anda. Data ini mencakup penyimpanan blok, penyimpanan objek, basis data, arsip, perangkat IoT, dan medium penyimpanan lain yang datanya dipertahankan. Melindungi data diam Anda dapat mengurangi risiko akses yang tidak sah, ketika enkripsi dan kontrol akses yang tepat diimplementasikan.

Enkripsi dan tokenisasi adalah dua skema perlindungan data yang berbeda tetapi sama pentingnya.

Tokenisasi adalah proses yang membuat Anda dapat menentukan token untuk merepresentasikan sebuah informasi sensitif (misalnya, token untuk merepresentasikan nomor kartu kredit pelanggan).

Token sendiri seharusnya tidak memiliki makna, dan tidak boleh didapatkan dari data yang ditokenisasi–oleh karenanya, digest kriptografis tidak dapat digunakan sebagai token. Dengan merencanakan pendekatan tokenisasi Anda secara saksama, Anda dapat memberikan perlindungan tambahan untuk konten Anda, dan Anda dapat memastikan bahwa Anda memenuhi persyaratan kepatuhan. Sebagai contoh, Anda dapat mempersempit cakupan kepatuhan sistem pemrosesan kartu kredit jika Anda memanfaatkan token, bukan nomor kartu kredit.

Enkripsi adalah cara mentransformasi konten dengan cara yang membuatnya tidak dapat dibaca tanpa menggunakan kunci rahasia yang diperlukan untuk mendekripsi konten agar kembali menjadi plaintext. Baik tokenisasi maupun enkripsi dapat digunakan untuk mengamankan dan melindungi informasi sebagaimana semestinya. Selain itu, masking adalah teknik yang memungkinkan bagian data diredaksi hingga data yang tersisa tidak lagi dianggap sensitif. Misalnya, PCI-DSS memungkinkan empat digit terakhir dari nomor kartu dipertahankan di luar batasan cakupan kepatuhan untuk pembuatan indeks.

Lakukan audit penggunaan kunci enkripsi: Pastikan bahwa Anda memahami dan mengaudit penggunaan kunci enkripsi guna memvalidasi bahwa mekanisme kontrol akses pada kunci diimplementasikan dengan tepat. Sebagai contoh, setiap layanan AWS yang menggunakan kunci AWS KMS mencatat setiap log penggunaan di AWS CloudTrail. Anda selanjutnya dapat membuat kueri AWS CloudTrail, dengan menggunakan alat seperti Wawasan Amazon CloudWatch, guna memastikan bahwa semua penggunaan kunci Anda valid.

Praktik terbaik

- SEC08-BP01 Mengimplementasikan manajemen kunci yang aman
- SEC08-BP02 Menerapkan enkripsi data diam
- SEC08-BP03 Mengotomatiskan perlindungan data diam
- SEC08-BP04 Menerapkan kontrol akses
- SEC08-BP05 Menggunakan mekanisme untuk mencegah orang mengakses data

SEC08-BP01 Mengimplementasikan manajemen kunci yang aman

Manajemen kunci yang aman mencakup penyimpanan, rotasi, kontrol akses, dan pemantauan materi kunci yang diperlukan untuk mengamankan data diam untuk beban kerja Anda.

Hasil yang diinginkan: Mekanisme manajemen kunci yang dapat diskalakan, dapat diulang, dan dapat diotomatiskan. Mekanisme ini harus memberikan kemampuan untuk menerapkan hak

akses paling rendah ke materi kunci, memberikan keseimbangan yang tepat antara ketersediaan, kerahasiaan, dan integritas kunci. Akses ke kunci harus dipantau, dan materi kunci dirotasi melalui proses otomatis. Materi kunci tidak boleh diakses oleh identitas manusia.

Antipola umum:

- Akses manusia ke materi kunci yang tidak terenkripsi.
- Membuat algoritma kriptografi kustom.
- Izin yang terlalu luas untuk mengakses materi kunci.

Manfaat menjalankan praktik terbaik ini: Dengan membuat mekanisme manajemen kunci yang aman untuk beban kerja Anda, Anda dapat membantu memberikan perlindungan untuk konten Anda dari akses yang tidak sah. Selain itu, Anda mungkin harus mematuhi persyaratan peraturan untuk mengenkripsi data Anda. Solusi manajemen kunci yang efektif dapat memberikan mekanisme teknis yang selaras dengan peraturan tersebut untuk melindungi materi kunci.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Banyak persyaratan peraturan dan praktik terbaik yang menyertakan enkripsi data diam sebagai kontrol keamanan mendasar. Untuk mematuhi kontrol ini, beban kerja Anda memerlukan mekanisme untuk menyimpan dan mengelola materi kunci yang digunakan untuk mengenkripsi data diam Anda.

AWS menawarkan AWS Key Management Service (AWS KMS) untuk menyediakan penyimpanan yang tahan lama, aman, dan redundan untuk kunci AWS KMS. Banyak layanan AWS terintegrasi dengan AWS KMS untuk mendukung enkripsi data Anda. AWS KMS menggunakan modul keamanan perangkat keras yang divalidasi FIPS 140-2 Level 3 untuk melindungi kunci Anda. Tidak ada mekanisme untuk mengekspor kunci AWS KMS ke dalam bentuk teks biasa.

Saat men-deploy beban kerja menggunakan strategi multiakun, merupakan salah satu praktik terbaik untuk menyimpan kunci AWS KMS di akun yang sama dengan beban kerja yang menggunakannya. Dalam model terdistribusi ini, tanggung jawab untuk mengelola kunci AWS KMS diemban oleh tim aplikasi. Dalam kasus penggunaan lainnya, organisasi dapat memilih untuk menyimpan kunci AWS KMS ke dalam sebuah akun terpusat. Struktur terpusat ini memerlukan kebijakan tambahan untuk mengaktifkan akses lintas akun yang diperlukan agar akun beban kerja dapat mengakses kunci yang disimpan di akun terpusat tersebut, tetapi mungkin lebih ideal untuk kasus penggunaan di mana satu kunci digunakan bersama-sama di beberapa Akun AWS.

Terlepas dari lokasi penyimpanan materi kunci, akses ke kunci harus dikontrol dengan ketat melalui penggunaan kebijakan kunci dan kebijakan IAM Kebijakan kunci adalah cara utama untuk mengontrol akses ke kunci AWS KMS. Selain itu, pemberian kunci AWS KMS dapat memberikan akses ke layanan AWS untuk mengenkripsi dan mendekripsi data atas nama Anda. Luangkan waktu untuk mempelajari praktik terbaik untuk kontrol akses ke kunci AWS KMS Anda.

Salah satu praktik terbaik adalah memantau penggunaan kunci enkripsi untuk mendeteksi pola akses yang tidak biasa. Operasi yang dijalankan menggunakan kunci yang dikelola AWS dan kunci yang dikelola pelanggan yang disimpan AWS KMS dapat dicatatkan dalam log di AWS CloudTrail dan harus ditinjau secara berkala. Perhatian khusus harus diberikan pada pemantauan peristiwa penghancuran kunci. Untuk mengurangi penghancuran materi kunci yang tidak disengaja atau berbahaya, peristiwa penghancuran kunci tidak langsung menghapus materi kunci. Upaya untuk menghapus kunci di AWS KMS tunduk pada masa tunggu, yakni secara default 30 hari, sehingga memberikan waktu kepada administrator untuk meninjau tindakan ini dan membatalkan permintaan jika perlu.

Sebagian besar layanan AWS menggunakan AWS KMS secara transparan bagi Anda - satu-satunya persyaratan Anda adalah memutuskan apakah akan menggunakan kunci yang dikelola AWS atau dikelola pelanggan. Jika beban kerja Anda memerlukan penggunaan langsung AWS KMS untuk mengenkripsi atau mendekripsi data, praktik terbaiknya adalah menggunakan enkripsi amplop untuk melindungi data Anda. Perintah SDK Enkripsi AWS dapat menyediakan primitive enkripsi sisi klien aplikasi Anda untuk mengimplementasikan enkripsi amplop dan terintegrasi dengan AWS KMS.

Langkah implementasi

1. Tentukan opsi manajemen kunci (yang dikelola AWS atau dikelola pelanggan) yang tepat untuk kunci Anda.
    - Untuk memudahkan penggunaan, AWS menawarkan kunci yang dimiliki AWS dan yang dikelola AWS untuk sebagian besar layanan, yang menyediakan kemampuan enkripsi data diam tanpa perlu mengelola materi kunci atau kebijakan kunci.
    - Saat menggunakan kunci yang dikelola pelanggan, pertimbangkan penyimpanan kunci default untuk memberikan keseimbangan terbaik antara ketangkasan, keamanan, kedaulatan data, dan ketersediaan. Kasus-kasus penggunaan lain mungkin memerlukan penggunaan penyimpanan kunci kustom dengan AWS CloudHSM atau penyimpanan kunci eksternal.
2. Tinjau daftar layanan yang sedang Anda gunakan untuk beban kerja Anda untuk memahami bagaimana AWS KMS terintegrasi dengan layanan. Misalnya, instans EC2 dapat menggunakan volume EBS terenkripsi, yang memverifikasi bahwa snapshot Amazon EBS yang dibuat dari

volume tersebut juga dienkripsi menggunakan kunci yang dikelola pelanggan dan mengurangi pengungkapan data snapshot yang tidak terenkripsi secara tidak disengaja.

- Bagaimana layanan AWS menggunakan AWS KMS
- Untuk informasi mendetail tentang opsi enkripsi yang ditawarkan oleh layanan AWS, lihat topik Enkripsi Data Diam di panduan pengguna atau panduan developer untuk layanan tersebut.

3. Implementasikan AWS KMS: AWS KMS memudahkan Anda membuat dan mengelola kunci serta mengontrol penggunaan enkripsi di berbagai layanan AWS dan dalam aplikasi Anda.
   - Memulai: AWS Key Management Service (AWS KMS)
   - Tinjau praktik terbaik untuk kontrol akses ke kunci AWS KMS Anda.
4. Pertimbangkan AWS Encryption SDK: Gunakan AWS Encryption SDK dengan integrasi AWS KMS ketika aplikasi Anda harus mengenkripsi data di sisi klien.
   - AWS Encryption SDK
5. Aktifkan IAM Access Analyzer agar secara otomatis meninjau dan memberi tahu jika ada kebijakan kunci AWS KMS yang terlalu luas.
6. Aktifkan Security Hub agar menerima notifikasi jika ada kebijakan kunci yang salah konfigurasi, kunci yang dijadwalkan untuk dihapus, atau kunci tanpa pengaktifan rotasi otomatis.
7. Tentukan tingkat pencatatan log yang sesuai untuk kunci AWS KMS Anda. Karena panggilan ke AWS KMS, termasuk peristiwa hanya-baca, dicatat ke log, jumlah log CloudTrail yang terkait dengan AWS KMS bisa jadi sangat banyak.
   - Beberapa organisasi lebih suka memisahkan aktivitas pencatatan log AWS KMS ke dalam jalur terpisah. Untuk detail selengkapnya, lihat bagian Mencatatkan log panggilan API AWS KMS dengan CloudTrail dalam panduan AWS KMS untuk developer.

Sumber daya

Dokumen terkait:

- AWS Key Management Service
- Layanan dan alat kriptografi AWS
- Melindungi Data Amazon S3 Menggunakan Enkripsi
- Enkripsi amplop
- Janji kedaulatan digital
- Menjelaskan operasi kunci AWS KMS, membawa kunci Anda sendiri, penyimpanan kunci kustom, dan portabilitas teks sandi

- Detail kriptografi AWS Key Management Service

Video terkait:

- Cara Kerja Enkripsi di AWS
- Mengamankan Penyimpanan Blok di AWS
- Perlindungan data AWS: Menggunakan gembok, kunci, tanda tangan, dan sertifikat

Contoh terkait:

- Mengimplementasikan mekanisme kontrol akses lanjutan menggunakan AWS KMS

SEC08-BP02 Menerapkan enkripsi data diam

Anda harus menerapkan penggunaan enkripsi untuk data diam. Enkripsi menjaga kerahasiaan data sensitif jika terjadi akses tidak sah atau pengungkapan yang tidak disengaja.

Hasil yang diinginkan: Data pribadi harus dienkripsi secara default saat diam. Enkripsi membantu menjaga kerahasiaan data dan memberikan lapisan perlindungan tambahan terhadap pengungkapan atau eksfiltrasi data yang disengaja atau tidak disengaja. Data yang dienkripsi tidak dapat dibaca atau diakses tanpa membuka enkripsi data terlebih dahulu. Semua data tersimpan yang tidak dienkripsi harus diinventarisasi dan dikontrol.

Antipola umum:

- Tidak menggunakan konfigurasi enkripsikan secara default.
- Memberikan akses yang terlalu permisif ke kunci dekripsi.
- Tidak memantau penggunaan kunci enkripsi dan dekripsi.
- Menyimpan data tidak terenkripsi.
- Menggunakan kunci enkripsi yang sama untuk semua data tanpa memperhatikan penggunaan, jenis, dan klasifikasi data.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Petakan kunci enkripsi ke klasifikasi data dalam beban kerja Anda. Pendekatan ini membantu melindungi dari akses yang terlalu permisif saat menggunakan kunci enkripsi tunggal atau yang sangat kecil untuk data Anda (lihat SEC07-BP01 Mengidentifikasi data dalam beban kerja Anda).

AWS Key Management Service (AWS KMS) terintegrasi dengan berbagai layanan AWS untuk mempermudah enkripsi data diam. Misalnya, di Amazon Simple Storage Service (Amazon S3), Anda dapat mengatur enkripsi default pada bucket agar semua objek baru dienkripsi secara otomatis. Saat menggunakan AWS KMS, pertimbangkan seberapa ketat pembatasan data yang perlu dilakukan. Kunci AWS KMS default dan yang dikontrol layanan dikelola dan digunakan atas nama Anda oleh AWS. Untuk data sensitif yang memerlukan akses terperinci ke kunci enkripsi yang mendasarinya, pertimbangkan kunci yang dikelola pelanggan (CMK). Anda memiliki kontrol penuh atas CMK, termasuk rotasi dan manajemen akses melalui penggunaan kebijakan kunci.

Selain itu, Amazon Elastic Compute Cloud (Amazon EC2) dan Amazon S3 mendukung penerapan enkripsi dengan mengatur enkripsi default. Anda dapat menggunakan Aturan AWS Config untuk memeriksa secara otomatis apakah Anda sedang menggunakan enkripsi. Misalnya, untuk volume Amazon Elastic Block Store (Amazon EBS), instans Amazon Relational Database Service (Amazon RDS), dan bucket Amazon S3.

AWS juga menyediakan opsi untuk enkripsi sisi klien, sehingga Anda dapat mengenkripsi data sebelum mengunggahnya ke cloud. AWS Encryption SDK menyediakan cara untuk mengenkripsi data Anda menggunakan enkripsi amplop. Anda memberikan kunci pembungkus, dan AWS Encryption SDK menghasilkan kunci data unik untuk setiap objek data yang dienkripsi. Pertimbangkan AWS CloudHSM jika Anda memerlukan modul keamanan perangkat keras (HSM) penyewa tunggal terkelola. AWS CloudHSM memungkinkan Anda membuat, mengimpor, dan mengelola kunci kriptografi pada HSM tervalidasi FIPS 140-2 level 3. Beberapa kasus penggunaan AWS CloudHSM termasuk melindungi kunci pribadi untuk menerbitkan otoritas sertifikat (CA), dan mengaktifkan enkripsi data transparan (TDE) untuk basis data Oracle. SDK Klien AWS CloudHSM menyediakan perangkat lunak yang dapat Anda gunakan untuk mengenkripsi data sisi klien menggunakan kunci yang disimpan di dalam AWS CloudHSM sebelum mengunggah data Anda ke AWS. Amazon DynamoDB Encryption Client juga mendukung enkripsi dan penandatanganan item sebelum diunggah ke tabel DynamoDB.

Langkah implementasi

- Terapkan enkripsi data diam untuk Amazon S3: Implementasikan enkripsi default bucket Amazon S3.

Konfigurasikan enkripsi default untuk volume Amazon EBS baru: Tentukan bahwa Anda ingin membuat semua volume Amazon EBS baru dalam bentuk terenkripsi, dengan opsi penggunaan kunci default yang disediakan oleh AWS, atau kunci yang Anda buat.

Konfigurasikan Amazon Machine Image (AMI) terenkripsi: Menyalin AMI yang ada dengan enkripsi aktif akan mengenkripsi volume root dan snapshot secara otomatis.

Konfigurasikan enkripsi Amazon RDS: Konfigurasikan enkripsi untuk klaster dan snapshot basis data Amazon RDS Anda saat diam menggunakan opsi enkripsi.

Buat dan konfigurasikan kunci AWS KMS dengan kebijakan yang membatasi akses ke pengguna utama yang sesuai untuk setiap klasifikasi data: Misalnya, buat satu kunci AWS KMS untuk mengenkripsi data produksi dan satu kunci untuk mengenkripsi data pengembangan atau pengujian. Anda juga dapat menyediakan kunci akses ke Akun AWS lainnya. Pertimbangkan untuk memiliki akun yang berbeda untuk lingkungan pengembangan dan produksi Anda. Jika lingkungan produksi Anda perlu mendekripsi artefak di akun pengembangan, Anda dapat mengedit kebijakan CMK yang digunakan untuk mengenkripsi artefak pengembangan agar akun produksi dapat mendekripsi artefak tersebut. Kemudian lingkungan produksi dapat menyerap data yang didekripsi untuk digunakan dalam produksi.

Konfigurasikan enkripsi di layanan AWS tambahan: Untuk layanan AWS lain yang Anda gunakan, lihat dokumentasi keamanan untuk layanan terkait guna menentukan opsi enkripsi untuk layanan tersebut.

Sumber daya

Dokumen terkait:

- AWS Crypto Tools
- Dokumentasi AWS
- AWS Encryption SDK
- Laporan Resmi Detail Kriptografi AWS KMS
- AWS Key Management Service
- Layanan dan alat kriptografi AWS
- Enkripsi Amazon EBS
- Enkripsi default untuk volume Amazon EBS

- Mengenkripsi Sumber Daya Amazon RDS
- Bagaimana cara mengaktifkan enkripsi default untuk bucket Amazon S3?
- Melindungi Data Amazon S3 Menggunakan Enkripsi

Video terkait:

- Cara Kerja Enkripsi di AWS
- Mengamankan Penyimpanan Blok di AWS

SEC08-BP03 Mengotomatiskan perlindungan data diam

Gunakan alat otomatis untuk memvalidasi dan menegakkan kontrol data diam secara terus menerus, misalnya, memastikan bahwa hanya ada sumber daya penyimpanan terenkripsi. Anda bisa mengotomatiskan validasi bahwa semua volume EBS telah dienkripsi menggunakan Aturan AWS Config. AWS Security Hub juga dapat memverifikasi beberapa kontrol yang berbeda melalui pemeriksaan otomatis berdasarkan standar keamanan. Selain itu, Aturan AWS Config Anda secara otomatis bisa memperbaiki sumber daya yang tidak patuh.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Data diam mewakili data yang Anda pertahankan di penyimpanan non-volatile selama durasi apa pun di beban kerja Anda. Data ini mencakup penyimpanan blok, penyimpanan objek, basis data, arsip, perangkat IoT, dan medium penyimpanan lain di mana datanya dipertahankan. Melindungi data diam Anda dapat mengurangi risiko akses yang tidak sah, ketika enkripsi dan kontrol akses yang tepat diimplementasikan.

Terapkan enkripsi data diam: Anda harus memastikan bahwa satu-satunya cara untuk menyimpan data adalah dengan menggunakan enkripsi. AWS KMS terintegrasi secara mulus dengan beberapa layanan AWS untuk mempermudah Anda mengenkripsi semua data diam Anda. Misalnya, di Amazon Simple Storage Service (Amazon S3) Anda bisa mengatur enkripsi default pada bucket sehingga semua objek baru terenkripsi secara otomatis. Selain itu, Amazon EC2 dan Amazon S3 mendukung penegakan enkripsi dengan mengatur enkripsi default. Anda dapat menggunakan AWS Managed Config Rules untuk memeriksa secara otomatis bahwa Anda menggunakan enkripsi, misalnya, untuk volume EBS, instans Amazon Relational Database Service (Amazon RDS), dan bucket Amazon S3.

Sumber daya

Dokumen terkait:

- AWS Crypto Tools
- AWS Encryption SDK

Video terkait:

- Cara Kerja Enkripsi di AWS
- Mengamankan Penyimpanan Blok di AWS

SEC08-BP04 Menerapkan kontrol akses

Untuk membantu melindungi data diam, terapkan kontrol akses menggunakan mekanisme, seperti isolasi dan versioning, serta terapkan prinsip hak akses paling rendah. Cegah pemberian akses publik ke data Anda.

Hasil yang diinginkan: Memastikan hanya pengguna yang sah yang dapat mengakses data sesuai kebutuhan. Melindungi data Anda dengan pencadangan dan versioning rutin untuk mencegah pengubahan atau penghapusan data yang disengaja atau tidak disengaja. Mengisolasi data penting dari data lain untuk melindungi kerahasiaan dan integritas data tersebut.

Antipola umum:

- Menyimpan data dengan kebutuhan atau klasifikasi sensitivitas yang berbeda secara bersamaan.
- Menggunakan izin yang terlalu permisif pada kunci dekripsi.
- Salah mengklasifikasi data.
- Tidak menyimpan pencadangan terperinci untuk data penting.
- Memberikan akses terus-menerus ke data produksi.
- Tidak mengaudit akses data atau meninjau izin secara rutin

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Rendah

Panduan implementasi

Beberapa kontrol dapat membantu melindungi data diam, termasuk akses (menggunakan hak akses paling rendah), isolasi, dan versioning. Akses ke data Anda harus diaudit menggunakan mekanisme deteksi, seperti AWS CloudTrail, dan log tingkat layanan, seperti log akses Amazon Simple Storage Service (Amazon S3). Anda harus membuat daftar data mana yang dapat diakses publik, dan menyusun rencana untuk mengurangi jumlah data yang tersedia untuk publik dari waktu ke waktu.

Kunci Vault Amazon S3 Glacier dan Kunci Objek Amazon S3 memberikan kontrol akses wajib untuk objek di Amazon S3—begitu kebijakan vault dikunci dengan opsi kepatuhan, kebijakan tersebut tidak akan dapat diubah hingga kedaluwarsa, bahkan oleh pengguna root sekalipun.

Langkah implementasi

- Terapkan akses kontrol: Terapkan akses kontrol dengan hak akses paling rendah, termasuk akses ke kunci enkripsi.
- Pisahkan data berdasarkan tingkat klasifikasi yang berbeda: Gunakan berbagai Akun AWS untuk tingkat klasifikasi, dan kelola akun tersebut menggunakan AWS Organizations.
- Tinjau kebijakan AWS Key Management Service (AWS KMS): Tinjau tingkat akses yang diberikan di kebijakan AWS KMS.
- Tinjau izin objek dan bucket Amazon S3: Tinjau tingkat akses yang diberikan dalam kebijakan bucket S3 secara rutin. Praktik terbaiknya adalah menghindari penggunaan bucket yang dapat dibaca atau ditulis oleh publik. Coba gunakan AWS Config untuk mendeteksi bucket yang tersedia untuk publik, dan Amazon CloudFront untuk menyajikan konten dari Amazon S3. Pastikan bucket yang seharusnya tidak mengizinkan akses publik telah dikonfigurasi dengan benar untuk mencegah akses publik. Secara default, semua bucket S3 bersifat privat, dan hanya dapat diakses oleh pengguna yang telah diberi akses secara eksplisit.
- Aktifkan AWS IAM Access Analyzer: IAM Access Analyzer menganalisis bucket Amazon S3 dan menghasilkan temuan saat kebijakan S3 memberikan izin ke entitas eksternal.
- Aktifkan versioning Amazon S3 dan kunci objek jika perlu.
- Gunakan Inventaris Amazon S3: Inventaris Amazon S3 dapat digunakan untuk mengaudit serta melaporkan replikasi dan status enkripsi objek S3.
- Tinjau izin Amazon EBS dan berbagi AMI: Dengan izin berbagi, Anda dapat membagikan gambar dan volume kepada Akun AWS eksternal ke beban kerja Anda.
- Tinjau AWS Resource Access Manager: Berbagi secara berkala untuk menentukan apakah sumber daya harus terus dibagikan. Dengan Resource Access Manager, Anda dapat membagikan

sumber daya seperti kebijakan AWS Network Firewall, aturan Amazon Route 53 Resolver, dan subnet dalam Amazon VPC Anda. Audit sumber daya yang dibagikan secara rutin dan hentikan pembagian sumber daya yang sudah tidak perlu dibagikan.

Sumber daya

Praktik Terbaik Terkait:

- SEC03-BP01 Menetapkan persyaratan akses
- SEC03-BP02 Memberikan hak akses paling rendah

Dokumen terkait:

- Laporan Resmi Detail Kriptografi AWS KMS
- Pengantar Manajemen Izin Akses ke Sumber Daya Amazon S3
- Gambaran umum manajemen akses ke sumber daya AWS KMS Anda
- Aturan AWS Config
- Amazon S3 + Amazon CloudFront: Kombinasi Fantastis di Cloud
- Menggunakan versioning
- Mengunci Objek Menggunakan Kunci Objek Amazon S3
- Membagikan Snapshot Amazon EBS
- AMI Bersama
- Meng-hosting aplikasi satu halaman aktif Amazon S3

Video terkait:

- Mengamankan Penyimpanan Blok di AWS

SEC08-BP05 Menggunakan mekanisme untuk mencegah orang mengakses data

Cegah semua pengguna dari mengakses sistem dan data sensitif secara langsung saat kondisi operasional normal. Misalnya, gunakan alur kerja manajemen perubahan untuk mengelola instans Amazon Elastic Compute Cloud (Amazon EC2) menggunakan alat, bukan dengan mengizinkan

akses langsung atau host bastion. Hal ini dapat dicapai dengan AWS Systems Manager Automation, yaitu menggunakan dokumen otomatis yang berisi langkah yang Anda gunakan untuk menjalankan tugas. Dokumen tersebut dapat disimpan di kontrol sumber, ditinjau oleh rekan sebelum dijalankan, dan diuji secara menyeluruh untuk meminimalkan risiko dibandingkan akses shell. Pengguna bisnis dapat menggunakan dasbor, bukan akses langsung ke penyimpanan data, untuk menjalankan kueri. Ketika pipeline CI/CD tidak digunakan, tentukan proses dan kontrol mana yang diperlukan agar dapat menyediakan mekanisme akses break-glass nonaktif secara normal.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

- Implementasikan mekanisme untuk mencegah orang dari mengakses data: Mekanisme termasuk menggunakan dasbor, seperti Amazon QuickSight, untuk menampilkan data ke pengguna ketimbang membuat kueri secara langsung.
  - Amazon QuickSight
- Otomatiskan manajemen konfigurasi: Lakukan tindakan dari jarak jauh, terapkan dan validasikan konfigurasi keamanan secara otomatis menggunakan layanan atau alat manajemen konfigurasi. Hindari penggunaan host bastion atau mengakses instans EC2 secara langsung.
  - AWS Systems Manager
  - AWS CloudFormation
  - Pipeline CI/CD untuk templat AWS CloudFormation di AWS

Sumber daya

Dokumen terkait:

- Laporan Resmi Detail Kriptografi AWS KMS

Video terkait:

- How Encryption Works in AWS
- Securing Your Block Storage on AWS

Melindungi data bergerak

Data bergerak adalah data yang dikirimkan dari satu sistem ke sistem lainnya. Ini mencakup komunikasi antarsumber daya di dalam beban kerja Anda juga komunikasi antara layanan lain dan pengguna akhir Anda. Dengan menyediakan tingkat perlindungan yang tepat untuk data bergerak, kerahasiaan dan integritas data di beban kerja Anda terlindungi.

Amankan data di antara VPC atau lokasi on-premise: Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi jaringan yang aman dan bersifat privat antara konektivitas on-premise atau Amazon Virtual Private Cloud (Amazon VPC) ke layanan yang di-hosting di AWS. Anda dapat mengakses layanan AWS, layanan pihak ketiga, dan layanan di Akun AWS lainnya seolah layanan tersebut berada di jaringan privat Anda. Dengan AWS PrivateLink, Anda dapat mengakses layanan lintas akun dengan CIDR IP yang tumpang tindih tanpa memerlukan Gateway Internet atau NAT. Anda juga tidak harus mengonfigurasikan aturan firewall, definisi jalur, atau tabel rute. Lalu lintas tetap berada di backbone Amazon dan tidak berjalan di internet, sehingga data Anda tetap terlindungi. Anda dapat mempertahankan kepatuhan dengan regulasi kepatuhan khusus industri, seperti HIPAA dan Perlindungan Privasi Uni Eropa/AS (EU/US Privacy Shield). AWS PrivateLink mudah bekerja dengan solusi pihak ketiga untuk membuat jaringan global yang disederhanakan, sehingga Anda dapat mempercepat migrasi ke cloud dan mengambil keuntungan dari layanan AWS yang tersedia.

Praktik terbaik

- SEC09-BP01 Mengimplementasikan manajemen sertifikat dan kunci keamanan
- SEC09-BP02 Menerapkan enkripsi data bergerak
- SEC09-BP03 Mengotomatiskan deteksi akses data yang tidak dimaksudkan
- SEC09-BP04 Sahkan komunikasi jaringan

SEC09-BP01 Mengimplementasikan manajemen sertifikat dan kunci keamanan

Sertifikat Keamanan Lapisan Pengangkutan (TLS) digunakan untuk mengamankan komunikasi jaringan dan menetapkan identitas situs web, sumber daya, dan beban kerja di internet, serta jaringan privat.

Hasil yang diinginkan: Sistem manajemen sertifikat aman yang dapat menyediakan, men-deploy, menyimpan, dan memperpanjang sertifikat di dalam infrastruktur kunci publik (PKI). Mekanisme

manajemen kunci dan sertifikat yang aman mencegah pengungkapan materi kunci privat sertifikat dan secara otomatis memperpanjang sertifikat secara berkala. Mekanisme ini juga terintegrasi dengan layanan lain untuk menyediakan komunikasi jaringan yang aman dan identitas untuk sumber daya mesin di dalam beban kerja Anda. Materi kunci tidak boleh diakses oleh identitas manusia.

Antipola umum:

- Melakukan langkah-langkah manual selama proses deployment atau perpanjangan sertifikat.
- Kurang memperhatikan hierarki otoritas sertifikat (CA) saat merancang CA privat.
- Menggunakan sertifikat yang ditandatangani sendiri untuk sumber daya publik.

Manfaat menjalankan praktik terbaik ini:

- Sederhanakan manajemen sertifikat melalui deployment dan perpanjangan otomatis
- Dorong enkripsi data bergerak menggunakan sertifikat TLS
- Peningkatan keamanan dan auditabilitas tindakan sertifikat yang dilakukan oleh otoritas sertifikat
- Manajemen tugas-tugas manajemen di berbagai lapisan hierarki CA

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Beban kerja modern banyak memanfaatkan komunikasi jaringan terenkripsi menggunakan protokol PKI seperti TLS. Manajemen sertifikat PKI mungkin kompleks, tetapi penyediaan, deployment, dan perpanjangan sertifikat secara otomatis dapat mengurangi gesekan yang berkaitan dengan manajemen sertifikat.

AWS menyediakan dua layanan untuk mengelola sertifikat PKI tujuan umum: AWS Certificate Manager dan AWS Private Certificate Authority (AWS Private CA). ACM adalah layanan primer yang digunakan oleh pelanggan untuk menyediakan, mengelola, dan melakukan deployment sertifikat untuk digunakan di beban kerja publik maupun AWS privat. ACM mengeluarkan sertifikat menggunakan AWS Private CA dan terintegrasi dengan banyak layanan terkelola AWS lainnya untuk menyediakan sertifikat TLS yang aman untuk beban kerja.

Dengan AWS Private CA, Anda dapat membuat otoritas sertifikat root atau subordinat Anda sendiri dan menerbitkan sertifikat TLS melalui API. Anda dapat menggunakan jenis-jenis sertifikat ini dalam skenario di mana Anda mengontrol dan mengelola rantai kepercayaan pada sisi klien koneksi TLS.

Selain kasus penggunaan TLS, AWS Private CA dapat digunakan untuk menerbitkan sertifikat ke pod Kubernetes, atestasi produk perangkat Matter, penandatanganan kode, dan kasus penggunaan lain dengan templat kustom. Anda juga dapat menggunakan IAM Roles Anywhere untuk memberikan kredensial IAM sementara ke beban kerja on-premise yang telah diberikan sertifikat X.509 yang ditandatangani oleh CA Privat Anda.

Selain ACM dan AWS Private CA, AWS IoT Core memberikan dukungan khusus untuk penyediaan, manajemen, dan deployment sertifikat PKI ke perangkat IoT. AWS IoT Core menyediakan mekanisme khusus untuk memasukkan perangkat IoT ke dalam infrastruktur kunci publik Anda dalam skala besar.

Pertimbangan untuk membangun hierarki CA privat

Ketika Anda perlu membuat CA privat, penting untuk berhati-hati dalam merancang hierarki CA dengan benar di awal. Salah satu praktik terbaiknya adalah men-deploy setiap tingkat hierarki CA Anda ke dalam Akun AWS yang terpisah saat membuat hierarki CA privat. Langkah sengaja ini mengurangi luas permukaan untuk setiap tingkat di dalam hierarki CA, sehingga mempermudah penemuan anomali dalam data log CloudTrail dan mengurangi ruang lingkup akses atau dampak jika terdapat akses tidak sah ke salah satu akun. CA root harus berada di akun terpisahnya sendiri dan hanya boleh digunakan untuk menerbitkan satu atau beberapa sertifikat CA perantara.

Kemudian, buat satu atau beberapa CA perantara di akun yang terpisah dari akun CA root untuk menerbitkan sertifikat bagi pengguna akhir, perangkat, atau beban kerja lainnya. Terakhir, terbitkan sertifikat dari CA root Anda ke CA perantara, yang pada gilirannya akan menerbitkan sertifikat kepada pengguna akhir atau perangkat Anda. Untuk informasi selengkapnya tentang perencanaan deployment CA dan perancangan hierarki CA, termasuk perencanaan ketahanan, replikasi lintas wilayah, berbagi CA di seluruh organisasi, dan lainnya, lihat Merencanakan deployment AWS Private CA Anda.

Langkah implementasi

1. Tentukan layanan AWS relevan yang diperlukan untuk kasus penggunaan Anda:
    - Banyak kasus penggunaan dapat memanfaatkan infrastruktur kunci publik AWS yang sudah ada dengan menggunakan AWS Certificate Manager. ACM dapat digunakan untuk melakukan deployment sertifikat TLS untuk server web, penyeimbang beban, atau penggunaan lain untuk sertifikat yang dipercaya secara publik.
    - Pertimbangkan AWS Private CA ketika Anda perlu membuat hierarki otoritas sertifikat privat Anda sendiri atau memerlukan akses ke sertifikat yang dapat diekspor. ACM kemudian dapat

digunakan untuk menerbitkan banyak jenis sertifikat entitas akhir menggunakan AWS Private CA.

- Untuk kasus penggunaan di mana sertifikat harus disediakan dalam skala besar untuk perangkat Internet untuk Segala (IoT) yang disematkan, pertimbangkan AWS IoT Core.

2. Implementasikan perpanjangan sertifikat otomatis jika memungkinkan:
   - Gunakan perpanjangan yang dikelola ACM untuk sertifikat yang diterbitkan oleh ACM bersama dengan layanan terkelola AWS yang terintegrasi.
3. Bangun jalur pencatatan dan audit:
   - Aktifkan log CloudTrail untuk melacak akses ke akun yang memiliki otoritas sertifikat. Pertimbangkan mengonfigurasi validasi integritas file log di CloudTrail untuk memverifikasi keaslian data log.
   - Buat dan tinjau secara berkala laporan audit yang mencantumkan sertifikat yang telah diterbitkan atau dicabut oleh CA privat Anda. Laporan ini dapat diekspor ke bucket S3.
   - Saat men-deploy CA pribadi, Anda juga perlu membuat bucket S3 untuk menyimpan Daftar Pencabutan Sertifikat (CRL). Untuk panduan mengonfigurasi bucket S3 ini berdasarkan persyaratan beban kerja Anda, lihat Merencanakan daftar pencabutan sertifikat (CRL).

Sumber daya

Praktik terbaik terkait:

- SEC02-BP02 Menggunakan kredensial sementara
- SEC08-BP01 Mengimplementasikan manajemen kunci yang aman
- SEC09-BP04 Sahkan komunikasi jaringan

Dokumen terkait:

- Cara meng-host dan mengelola seluruh infrastruktur sertifikat privat di AWS.
- Cara mengamankan hierarki CA Privat ACM skala korporasi untuk otomotif dan manufaktur
- Praktik terbaik CA privat
- Cara menggunakan AWS RAM untuk membagikan CA Privat ACM Anda lintas akun

Video terkait:

- Mengaktifkan CA Privat AWS Certificate Manager (lokakarya)

Contoh terkait:

- Lokakarya CA privat
- Lokakarya Manajemen Perangkat IOT (termasuk penyediaan perangkat)

Alat terkait:

- Plugin ke Kubernetes cert-manager untuk menggunakan AWS Private CA

SEC09-BP02 Menerapkan enkripsi data bergerak

Terapkan persyaratan enkripsi yang Anda tentukan berdasarkan kebijakan, kewajiban regulasi, dan standar organisasi Anda untuk membantu memenuhi persyaratan organisasi, hukum, dan kepatuhan. Hanya gunakan protokol yang dienkripsi ketika mengirimkan data sensitif di luar cloud privat virtual (VPC) Anda. Enkripsi membantu menjaga kerahasiaan data, bahkan ketika data berada di jaringan tidak tepercaya.

Hasil yang diinginkan: Semua data harus dienkripsi saat bergerak menggunakan protokol TLS dan rangkaian sandi yang aman. Lalu lintas jaringan antara sumber daya Anda dan internet harus dienkripsi untuk mengurangi akses tidak sah ke data. Lalu lintas jaringan yang hanya berada dalam lingkungan AWS internal Anda harus sebisa mungkin dienkripsi menggunakan TLS. Jaringan internal AWS dienkripsi secara default dan lalu lintas jaringan di dalam VPC tidak dapat dipalsukan atau dilacak kecuali pihak yang tidak sah telah memperoleh akses ke sumber daya yang menghasilkan lalu lintas (seperti instans Amazon EC2 dan kontainer Amazon ECS). Pertimbangkan untuk melindungi lalu lintas antarjaringan dengan jaringan privat virtual (VPN) IPsec.

Antipola umum:

- Menggunakan versi komponen SSL, TLS, rangkaian sandi yang tidak digunakan lagi (misalnya, SSL v3.0, kunci RSA 1024-bit, dan sandi RC4).
- Mengizinkan lalu lintas (HTTP) tidak terenkripsi ke atau dari sumber daya yang dapat diakses publik.
- Tidak memantau dan tidak mengganti sertifikat X.509 sebelum kedaluwarsa.
- Menggunakan sertifikat X.509 yang Anda buat sendiri untuk TLS.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Layanan AWS menyediakan titik akhir HTTPS menggunakan TLS untuk komunikasi, memberikan enkripsi data bergerak saat berkomunikasi dengan API AWS. Protokol yang tidak aman, seperti HTTP, dapat diaudit dan diblokir di VPC melalui penggunaan grup keamanan. Permintaan HTTP juga dapat secara otomatis diarahkan ke HTTPS di Amazon CloudFront atau pada Application Load Balancer. Anda memiliki kendali penuh atas sumber daya komputasi Anda untuk mengimplementasikan enkripsi data bergerak di seluruh layanan Anda. Selain itu, Anda dapat menggunakan sambungan VPN ke dalam VPC Anda dari jaringan eksternal atau AWS Direct Connect untuk mendukung enkripsi lalu lintas. Pastikan klien Anda melakukan panggilan ke API AWS menggunakan minimal TLS 1.2, karena AWS menghentikan penggunaan TLS 1.0 dan 1.1 pada Juni 2023. Jika Anda memiliki persyaratan khusus, tersedia solusi pihak ketiga di AWS Marketplace.

Langkah implementasi

- Terapkan enkripsi data bergerak: Persyaratan enkripsi yang Anda tetapkan harus didasarkan pada standar dan praktik terbaik terbaru dan hanya mengizinkan protokol yang aman. Misalnya, konfigurasikan grup keamanan untuk hanya mengizinkan protokol HTTPS ke penyeimbang beban aplikasi atau instans Amazon EC2.
- Konfigurasikan protokol yang aman di layanan edge: Konfigurasikan HTTPS dengan Amazon CloudFront dan gunakan profil keamanan yang sesuai dengan postur keamanan dan kasus penggunaan Anda.
- Gunakan VPN untuk koneksi eksternal: Pertimbangkan penggunaan VPN IPsec untuk mengamankan koneksi antartitik atau antarjaringan untuk membantu memberikan integritas dan privasi data.
- Konfigurasikan protokol yang aman di penyeimbang beban: Pilih kebijakan keamanan yang menyediakan rangkaian sandi terkuat yang didukung oleh klien yang akan terhubung ke pendengar. Membuat pendengar HTTPS untuk Application Load Balancer Anda.
- Konfigurasikan protokol yang aman di Amazon Redshift: Konfigurasikan klaster Anda untuk meminta koneksi lapisan soket aman (SSL) atau keamanan lapisan pengangkutan (TLS).
- Konfigurasikan protokol yang aman: Baca dokumentasi layanan AWS untuk menentukan kemampuan enkripsi data bergerak.
- Konfigurasikan akses yang aman saat melakukan pengunggahan ke bucket Amazon S3: Gunakan kontrol kebijakan bucket Amazon S3 untuk menerapkan akses aman ke data.

- Pertimbangkan penggunaan AWS Certificate Manager: ACM memungkinkan Anda untuk menyediakan, mengelola, dan melakukan deployment sertifikat TLS publik untuk digunakan dengan layanan AWS.
- Pertimbangkan penggunaan AWS Private Certificate Authority untuk kebutuhan PKI privat: AWS Private CA memungkinkan Anda membuat hierarki otoritas sertifikat (CA) pribadi untuk menerbitkan sertifikat X.509 entitas akhir yang dapat digunakan untuk membuat saluran TLS terenkripsi.

Sumber daya

Dokumen terkait:

- Dokumentasi AWS
- Menggunakan HTTPS dengan CloudFront
- Menghubungkan VPC ke jaringan jarak jauh menggunakan AWS Virtual Private Network
- Membuat pendengar HTTPS untuk Application Load Balancer Anda
- Tutorial: Mengonfigurasi SSL/TLS di Amazon Linux 2
- Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans DB
- Mengonfigurasi opsi-opsi keamanan untuk koneksi

SEC09-BP03 Mengotomatiskan deteksi akses data yang tidak dimaksudkan

Gunakan alat seperti Amazon GuardDuty untuk secara otomatis mendeteksi aktivitas mencurigakan atau mencoba memindahkan data di luar batas yang telah ditetapkan. Misalnya, GuardDuty dapat mendeteksi aktivitas baca Amazon Simple Storage Service (Amazon S3) yang tidak seperti biasanya dengan Temuan Exfiltration:S3/AnomalousBehavior. Selain GuardDuty, Log Alur Amazon VPCyang mendokumentasikan informasi lalu lintas jaringan, dapat digunakan dengan Amazon EventBridge untuk memicu deteksi koneksi tidak normal, baik yang berhasil maupun yang ditolak. Amazon S3 Access Analyzer dapat membantu mengukur data apa yang dapat diakses di dalam bucket Amazon S3 Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

- Otomatiskan deteksi akses data yang tidak dimaksudkan: Gunakan alat atau mekanisme deteksi untuk secara otomatis mendeteksi upaya untuk memindahkan data di luar batas yang ditetapkan; misalnya, untuk mendeteksi sistem basis data yang menyalin data ke host tidak dikenal.
  - Log Alur VPC
- Pertimbangkan Amazon Macie: Amazon Macie adalah layanan privasi data dan keamanan data terkelola secara penuh yang menggunakan machine learning dan pencocokan pola untuk menemukan dan melindungi data sensitif Anda di AWS.
  - Amazon Macie

Sumber daya

Dokumen terkait:

- Log Alur VPC
- Amazon Macie

SEC09-BP04 Sahkan komunikasi jaringan

Verifikasi identitas komunikasi dengan menggunakan protokol yang mendukung autentikasi, seperti Keamanan Lapisan Pengangkutan (TLS) atau IPsec.

Rancang beban kerja Anda untuk menggunakan protokol jaringan yang aman dan terautentikasi setiap kali berkomunikasi antara layanan, aplikasi, atau ke pengguna. Menggunakan protokol jaringan yang mendukung autentikasi dan otorisasi memberikan kontrol yang lebih kuat atas aliran jaringan dan mengurangi dampak akses yang tidak sah.

Hasil yang diinginkan: Beban kerja dengan arus lalu lintas bidang data dan bidang kontrol yang jelas antara layanan. Arus lalu lintas menggunakan protokol jaringan yang diautentikasi dan dienkripsi jika memungkinkan secara teknis.

Antipola umum:

- Alur lalu lintas yang tidak dienkripsi atau tidak diautentikasi dalam beban kerja Anda.
- Penggunaan kembali kredensial autentikasi oleh beberapa pengguna atau entitas.
- Hanya mengandalkan kontrol jaringan sebagai mekanisme kontrol akses.

- Membuat mekanisme autentikasi kustom, bukan mengandalkan mekanisme autentikasi standar industri.
- Alur lalu lintas yang terlalu permisif antara komponen layanan atau sumber daya lain di VPC.

Manfaat menjalankan praktik terbaik ini:

- Membatasi ruang lingkup dampak untuk akses tidak sah ke satu bagian dari beban kerja.
- Memberikan tingkat jaminan yang lebih tinggi bahwa tindakan hanya dilakukan oleh entitas yang diautentikasi.
- Meningkatkan pemisahan layanan dengan menggambarkan dengan jelas dan menerapkan antarmuka transfer data yang diinginkan.
- Meningkatkan pemantauan, pembuatan log, dan respons insiden melalui atribusi permintaan dan antarmuka komunikasi yang digambarkan dengan jelas.
- Memberikan pertahanan mendalam untuk beban kerja Anda dengan menggabungkan kontrol jaringan dengan kontrol autentikasi dan otorisasi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Rendah

Panduan implementasi

Pola lalu lintas jaringan beban kerja Anda dapat dikelompokkan ke dalam dua kategori:

- Lalu lintas timur-barat mewakili arus lalu lintas antara layanan yang membentuk beban kerja.
- Lalu lintas utara-selatan mewakili arus lalu lintas antara beban kerja Anda dan konsumen.

Meskipun enkripsi lalu lintas utara-selatan merupakan praktik umum, pengamanan lalu lintas timur-barat menggunakan protokol yang diautentikasi merupakan hal yang kurang umum. Praktik keamanan modern menyebutkan bahwa desain jaringan saja tidak cukup untuk memberikan hubungan yang dapat dipercaya antara dua entitas. Ketika dua layanan dapat berada dalam batas jaringan yang sama, sebaiknya enkripsi, autentikasi, dan otorisasi komunikasi di antara layanan-layanan tersebut tetap dilakukan.

Sebagai contoh, API layanan AWS menggunakan protokol tanda tangan Signature Version 4 (SIGv4) AWS untuk mengautentikasi pemanggil, dari jaringan mana pun permintaan tersebut berasal. Autentikasi ini memastikan bahwa API AWS dapat memverifikasi identitas yang meminta tindakan,

dan identitas tersebut kemudian dapat digabungkan dengan kebijakan untuk membuat keputusan otorisasi guna menentukan apakah tindakan tersebut harus diizinkan atau tidak.

Layanan seperti Amazon VPC Lattice dan Amazon API Gateway memungkinkan Anda menggunakan protokol tanda tangan SigV4 yang sama untuk menambahkan autentikasi dan otorisasi ke lalu lintas timur-barat dalam beban kerja Anda sendiri. Jika sumber daya di luar lingkungan AWS Anda perlu berkomunikasi dengan layanan yang memerlukan autentikasi dan otorisasi berbasis SigV4, Anda dapat menggunakan AWS Identity and Access Management (IAM) Roles Anywhere pada sumber daya non-AWS untuk memperoleh kredensial AWS sementara. Kredensial ini dapat digunakan untuk menandatangani permintaan ke layanan menggunakan SigV4 untuk memberi otorisasi akses.

Mekanisme umum lainnya untuk mengautentikasi lalu lintas timur-barat adalah autentikasi timbal balik TLS (mTLS). Banyak Internet untuk Segala (IoT), aplikasi bisnis-ke-bisnis, dan layanan mikro menggunakan mTLS untuk memvalidasi identitas kedua sisi komunikasi TLS melalui penggunaan sertifikat X.509 sisi klien dan server. Sertifikat ini dapat dikeluarkan oleh AWS Private Certificate Authority (AWS Private CA). Anda dapat menggunakan layanan seperti Amazon API Gateway dan AWS App Mesh untuk menyediakan autentikasi mTLS untuk komunikasi antar- atau intra-beban kerja. Meskipun mTLS menyediakan informasi autentikasi untuk kedua sisi komunikasi TLS, mekanisme untuk otorisasi tidak disediakan.

Akhirnya, OAuth 2.0 dan OpenID Connect (OIDC) adalah dua protokol yang biasanya digunakan untuk mengontrol akses ke layanan oleh pengguna, tetapi kini juga mulai populer untuk lalu lintas antar-layanan. API Gateway menyediakan pemberi otorisasi JSON Web Token (JWT), yang memungkinkan beban kerja membatasi akses ke rute API menggunakan JWT yang dikeluarkan dari penyedia identitas OIDC atau OAuth 2.0. Cakupan OAuth2 dapat digunakan sebagai sumber untuk keputusan otorisasi dasar, tetapi pemeriksaan otorisasi masih perlu diimplementasikan di lapisan aplikasi, dan cakupan OAuth2 saja tidak dapat mendukung kebutuhan otorisasi yang lebih kompleks.

Langkah implementasi

- Tentukan dan dokumentasikan alur jaringan beban kerja Anda: Langkah pertama dalam menerapkan strategi pertahanan mendalam adalah menentukan arus lalu lintas beban kerja Anda.
  - Buat diagram alur data yang secara jelas menggambarkan bagaimana data ditransmisikan antara berbagai layanan yang membentuk beban kerja Anda. Diagram ini merupakan langkah pertama untuk menerapkan alur tersebut melalui saluran jaringan yang diautentikasi.
  - Instrumentasikan beban kerja Anda dalam fase pengembangan dan pengujian untuk memvalidasi bahwa diagram alur data mencerminkan perilaku beban kerja secara akurat pada saat runtime.

    - Diagram alur data juga dapat berguna saat melakukan latihan pemodelan ancaman, seperti yang dijelaskan dalam SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi menggunakan model ancaman.
- Tetapkan kontrol jaringan: Pertimbangkan kemampuan AWS untuk menetapkan kontrol jaringan yang selaras dengan alur data Anda. Meskipun batas jaringan seharusnya tidak menjadi satu-satunya kontrol keamanan, batas tersebut menyediakan lapisan pada strategi pertahanan mendalam untuk melindungi beban kerja Anda.
    - Gunakan grup keamanan untuk menetapkan dan membatasi alur data antarsumber daya.
    - Pertimbangkan penggunaan AWS PrivateLink untuk berkomunikasi dengan layanan AWS dan layanan pihak ketiga yang mendukung AWS PrivateLink. Data yang dikirim melalui titik akhir antarmuka AWS PrivateLink tetap berada di dalam tulang punggung jaringan AWS dan tidak melintasi Internet publik.
- Implementasikan autentikasi dan otorisasi di seluruh layanan dalam beban kerja Anda: Pilih kumpulan layanan AWS yang paling tepat untuk menyediakan alur lalu lintas yang dienkripsi dan diautentikasi dalam beban kerja Anda.
    - Pertimbangkan Amazon VPC Lattice untuk mengamankan komunikasi antar layanan. VPC Lattice dapat menggunakan autentikasi SigV4 yang dikombinasikan dengan kebijakan autentikasi untuk mengontrol akses antar layanan.
    - Untuk komunikasi antar layanan menggunakan mTLS, pertimbangkan API Gateway atau App Mesh. AWS Private CA dapat digunakan untuk membuat hierarki CA pribadi yang mampu mengeluarkan sertifikat untuk digunakan dengan mTLS.
    - Saat mengintegrasikan dengan layanan menggunakan OAuth 2.0 atau OIDC, pertimbangkan API Gateway menggunakan pemberi otorisasi JWT.
    - Untuk komunikasi antara beban kerja Anda dan perangkat IoT, pertimbangkan AWS IoT Core, yang menyediakan beberapa opsi untuk enkripsi lalu lintas jaringan dan autentikasi.
- Pantau akses yang tidak sah: Terus pantau saluran komunikasi yang tidak diinginkan, pengguna utama tidak berwenang yang mencoba mengakses sumber daya yang dilindungi, dan pola akses yang tidak tepat lainnya.
    - Jika VPC Lattice digunakan untuk mengelola akses ke layanan Anda, pertimbangkan untuk mengaktifkan dan memantau log akses VPC Lattice. Log akses ini mencakup informasi tentang entitas yang meminta, informasi jaringan termasuk VPC sumber dan tujuan, dan metadata permintaan.
    - Pertimbangkan untuk mengaktifkan log alur VPC untuk menangkap metadata pada alur jaringan dan meninjau anomali secara berkala.

- Lihat Panduan Respons Insiden Keamanan AWS dan bagian Respons Insiden dari pilar keamanan Kerangka Kerja AWS Well-Architected untuk panduan lebih lanjut tentang merencanakan, menyimulasikan, dan menanggapi insiden keamanan.

Sumber daya

Praktik Terbaik Terkait:

- SEC03-BP07 Menganalisis akses lintas akun dan publik
- SEC02-BP02 Menggunakan kredensial sementara
- SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi menggunakan model ancaman

Dokumen terkait:

- Evaluating access control methods to secure Amazon API Gateway APIs
- Configuring mutual TLS authentication for a REST API
- How to secure API Gateway HTTP endpoints with JWT authorizer
- Authorizing direct calls to AWS services using AWS IoT Core credential provider
- Panduan Respons Insiden Keamanan AWS

Video terkait:

- AWS re:invent 2022: Introducing VPC Lattice
- AWS re:invent 2020: Serverless API authentication for HTTP APIs on AWS

Contoh terkait:

- Amazon VPC Lattice Workshop
- Zero-Trust Episode 1 – The Phantom Service Perimeter workshop

Respons insiden

Dengan kontrol deteksi dan preventif yang matang sekalipun, organisasi Anda harus mengimplementasikan mekanisme untuk merespons dan memitigasi potensi dampak insiden keamanan. Persiapan Anda sangat berpengaruh pada kemampuan tim Anda untuk beroperasi secara efektif selama insiden, untuk mengisolasi, membatasi, dan melakukan forensik terhadap masalah, serta untuk memulihkan operasi ke kondisi yang baik dan dikenal. Menetapkan alat dan akses sebelum terjadi insiden keamanan, lalu secara rutin melatih respons insiden melalui game day, membantu memastikan bahwa Anda dapat melakukan pemulihan sembari tetap meminimalkan gangguan bisnis.

Topik

Aspek respons insiden AWS

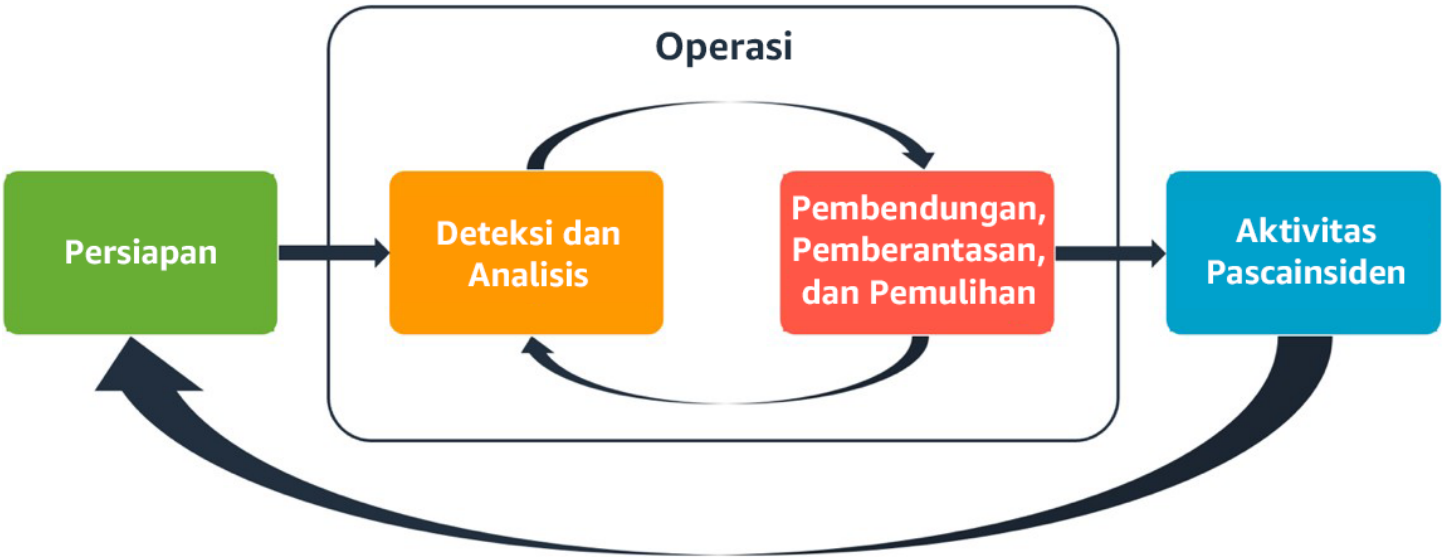
Semua pengguna AWS di dalam suatu organisasi harus memiliki pemahaman dasar tentang proses respons insiden keamanan, dan staf keamanan harus memahami cara merespons masalah keamanan. Pendidikan, pelatihan, dan pengalaman sangat penting untuk keberhasilan program respons insiden cloud dan idealnya diimplementasikan dengan baik sebelum kemungkinan insiden keamanan harus ditangani. Fondasi program respons insiden yang sukses di cloud adalah Persiapan, Operasi, dan Aktivitas Pascainsiden.

Untuk memahami masing-masing dari ketiga aspek ini, pelajari uraian berikut:

- Persiapan: Persiapkan tim respons insiden Anda untuk mendeteksi dan merespons insiden di dalam AWS dengan mengaktifkan kontrol detektif dan memverifikasi akses yang sesuai ke alat dan layanan cloud yang diperlukan. Selain itu, persiapkan playbook yang diperlukan, baik yang manual maupun otomatis, guna memastikan respons yang andal dan konsisten.
- Operasi: Lakukan operasi pada peristiwa keamanan dan potensi insiden dengan mengikuti fase respons insiden NIST: mendeteksi, menganalisis, mengendalikan, memberantas, dan memulihkan.

- Aktivitas Pascainsiden: Lakukan iterasi hasil peristiwa dan simulasi keamanan Anda untuk meningkatkan efisiensi respons Anda, meningkatkan nilai yang diperoleh dari respons dan investigasi, dan mengurangi risiko lebih lanjut. Anda harus belajar dari insiden dan bersikap proaktif untuk aktivitas perbaikan.

Diagram berikut ini menunjukkan alur aspek-aspek ini, selaras dengan siklus respons insiden NIST yang disebutkan sebelumnya, tetapi dengan operasi yang mencakup deteksi dan analisis dengan pengendalian, pemberantasan, dan pemulihan.



Aspek respons insiden AWS

Tujuan desain respons cloud

Meskipun proses dan mekanisme umum respons insiden, sebagaimana yang ditetapkan dalam NIST SP 800-61 Panduan Penanganan Insiden Keamanan Komputer, sudah benar, sebaiknya evaluasikan tujuan desain spesifik ini sesuai dengan respons insiden keamanan di lingkungan cloud:

- Tetapkan tujuan respons: Bekerja samalah dengan pemangku kepentingan, penasihat hukum, dan kepemimpinan organisasi untuk menentukan tujuan respons insiden. Beberapa tujuan umum antara lain mengendalikan dan memitigasi masalah, memulihkan sumber daya yang terdampak, mempertahankan data untuk keperluan forensik, memulihkan ke operasi yang diketahui aman, dan akhirnya memetik pelajaran dari insiden.
- Berikan respons dengan memanfaatkan cloud: Implementasikan pola respons di dalam cloud, tempat munculnya peristiwa dan data.

- Ketahui apa yang Anda miliki dan apa yang Anda butuhkan: Simpan log, sumber daya, snapshot, dan bukti lainnya dengan menyalin dan menyimpannya di akun cloud terpusat yang dibuat khusus untuk respons. Gunakan tag, metadata, dan mekanisme yang menerapkan kebijakan retensi. Anda harus memahami layanan apa yang Anda gunakan dan kemudian mengidentifikasi persyaratan untuk menyelidiki layanan tersebut. Untuk membantu memahami lingkungan Anda, Anda juga dapat menggunakan pemberian tag.
- Gunakan mekanisme deployment ulang: Jika kesalahan konfigurasi disebabkan oleh anomali keamanan, pemulihannya dapat semudah menghapus variansi melalui deployment ulang sumber daya dengan konfigurasi yang sesuai. Jika kemungkinan gangguan teridentifikasi, pastikan deployment ulang Anda mencakup mitigasi akar masalah yang berhasil dan diverifikasi.
- Lakukan otomatisasi jika memungkinkan: Jika masalah atau insiden muncul kembali, buat mekanisme yang secara terprogram membuat triase dan merespons peristiwa yang sama. Gunakan respons manusia untuk insiden yang unik, kompleks, atau sensitif yang tidak cukup ditangani oleh otomatisasi.
- Pilih solusi yang dapat diskalakan: Usahakan untuk menyesuaikan skalabilitas pendekatan organisasi Anda dengan komputasi cloud. Implementasikan mekanisme deteksi dan respons yang diskalakan di seluruh lingkungan Anda untuk secara efektif mengurangi waktu antara deteksi dan respons.
- Pelajari dan sempurnakan proses Anda: Jadilah proaktif dalam mengidentifikasi celah dalam proses, alat, atau personel Anda, dan implementasikan rencana untuk memperbaikinya. Simulasi merupakan metode yang aman untuk menemukan celah dan menyempurnakan proses.

Sasaran desain ini adalah pengingat untuk meninjau implementasi arsitektur Anda untuk kemampuan melakukan respons insiden dan deteksi ancaman. Saat Anda merencanakan implementasi cloud, pikirkan tentang respons terhadap suatu insiden, idealnya dengan metodologi respons yang baik secara forensik. Dalam beberapa kasus, ini berarti Anda mungkin memiliki beberapa organisasi, akun, dan alat yang disiapkan secara khusus untuk tugas-tugas respons ini. Alat dan fungsi tersebut harus tersedia untuk tim respons insiden melalui pipeline deployment. Sifatnya tidak boleh statis karena dapat menyebabkan risiko yang lebih besar.

Persiapan

Mempersiapkan insiden adalah hal yang sangat penting untuk respons insiden yang tepat waktu dan efektif. Persiapan dilakukan pada tiga domain:

- Personel: Mempersiapkan personel Anda untuk insiden keamanan melibatkan identifikasi pemangku kepentingan yang relevan untuk respons insiden dan melatih mereka tentang respons insiden dan teknologi cloud.
- Proses: Mempersiapkan proses untuk insiden keamanan melibatkan pendokumentasian arsitektur, pengembangan rencana respons insiden yang menyeluruh, dan pembuatan playbook untuk respons yang konsisten terhadap peristiwa keamanan.
- Teknologi: Mempersiapkan teknologi untuk insiden keamanan melibatkan penyiapan akses, agregasi dan pemantauan log yang diperlukan, implementasi mekanisme peringatan yang efektif, dan pengembangan respons serta kemampuan investigasi.

Setiap domain ini sama pentingnya untuk respons insiden yang efektif. Tidak ada program respons insiden yang lengkap atau efektif tanpa ketiga hal ini. Anda perlu mempersiapkan personel, proses, dan teknologi dengan integrasi yang erat agar siap menghadapi insiden.

Praktik terbaik

- SEC10-BP01 Identifikasikan sumber daya eksternal dan personel kunci
- SEC10-BP02 Membuat rencana manajemen insiden
- SEC10-BP03 Menyiapkan kemampuan forensik
- SEC10-BP04 Mengembangkan dan menguji playbook respons insiden keamanan
- SEC10-BP05 Menyediakan akses di awal
- SEC10-BP06 Melakukan deployment alat di awal
- SEC10-BP07 Menjalankan simulasi

SEC10-BP01 Identifikasikan sumber daya eksternal dan personel kunci

Identifikasikan kewajiban legal, sumber daya, dan personel internal serta eksternal yang dapat membantu organisasi Anda merespons insiden.

Saat Anda menentukan pendekatan Anda terhadap respons insiden di cloud, bersama dengan tim lainnya (seperti penasihat hukum, pimpinan, pemangku kepentingan bisnis, Layanan AWS Support, dan lainnya), Anda harus mengidentifikasi personel kunci, pemangku kepentingan, dan kontak yang relevan. Untuk mengurangi dependensi dan mempercepat waktu respons, pastikan tim Anda, tim keamanan spesialis, dan pemberi respons paham tentang layanan yang Anda gunakan dan memiliki kesempatan untuk praktik langsung.

Sebaiknya identifikasikan partner keamanan AWS eksternal yang dapat memberi Anda ahli dari luar perusahaan dan memberikan perspektif yang berbeda untuk melengkapi kemampuan respons Anda. Partner keamanan tepercaya Anda dapat membantu Anda mengidentifikasi potensi risiko atau ancaman yang belum Anda kenali dengan baik.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

- Identifikasi personel utama dalam organisasi Anda: Miliki daftar kontak personel di dalam organisasi Anda yang perlu Anda libatkan untuk merespons dan melakukan pemulihan dari insiden.
- Identifikasi partner eksternal: Bekerja samalah dengan partner eksternal yang dapat membantu Anda merespons dan melakukan pemulihan dari insiden, jika diperlukan.

Sumber daya

Dokumen terkait:

- Panduan Respons Insiden AWS

Video terkait:

- Prepare for and respond to security incidents in your AWS environment

Contoh terkait:

SEC10-BP02 Membuat rencana manajemen insiden

Dokumen pertama yang harus dikembangkan untuk merespons insiden adalah rencana respons insiden. Rencana respons insiden dirancang untuk menjadi landasan bagi program dan strategi respons insiden Anda.

Manfaat menjalankan praktik terbaik ini: Mengembangkan proses respons insiden yang menyeluruh dan jelas adalah kunci untuk program respons insiden yang sukses dan dapat diskalakan. Ketika peristiwa keamanan terjadi, langkah dan alur kerja yang jelas dapat membantu Anda merespons secara tepat waktu. Anda mungkin sudah memiliki proses respons insiden yang ada. Terlepas dari

status Anda saat ini, penting untuk memperbarui, mengiterasi, dan menguji proses respons insiden Anda secara rutin.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Rencana manajemen insiden sangat penting untuk merespons, memitigasi, dan pulih dari potensi dampak insiden keamanan. Rencana manajemen insiden adalah proses terstruktur untuk mengidentifikasi, memperbaiki, dan merespons insiden keamanan secara tepat waktu.

Cloud memiliki banyak peran dan persyaratan operasional yang sama yang juga ditemukan di lingkungan on-premise. Saat membuat rencana manajemen insiden, penting untuk mempertimbangkan strategi respons dan pemulihan yang paling selaras dengan hasil bisnis dan persyaratan kepatuhan Anda. Sebagai contoh, jika Anda mengoperasikan beban kerja di AWS yang patuh terhadap FedRAMP di Amerika Serikat, ada gunanya Anda mematuhi NIST SP 800-61 Panduan Penanganan Keamanan Komputer. Begitu juga, saat mengoperasikan beban kerja dengan data informasi pengenal pribadi (PII) Eropa, pertimbangkan skenario seperti cara Anda melindungi dan merespons permasalahan terkait residensi data yang diatur oleh Peraturan Perlindungan Data Umum (GDPR) Uni Eropa.

Saat membangun rencana manajemen insiden untuk beban kerja di AWS, mulailah dengan Model Tanggung Jawab Bersama AWS untuk membangun pendekatan pertahanan mendalam untuk respons insiden. Dalam model ini, AWS mengelola keamanan cloud, dan Anda bertanggung jawab atas keamanan di cloud. Ini artinya Anda mempertahankan kontrol dan bertanggung jawab atas kontrol keamanan yang ingin Anda implementasikan. Panduan Respons Insiden Keamanan AWS menguraikan konsep utama dan panduan mendasar untuk membangun rencana manajemen insiden berorientasi cloud.

Rencana manajemen insiden yang efektif harus diiterasi secara berkelanjutan, dan harus tetap mutakhir sesuai tujuan operasi cloud Anda. Pertimbangkan menggunakan rencana implementasi yang diuraikan di bawah seiring Anda membuat dan mengembangkan rencana manajemen insiden Anda.

Langkah implementasi

Tentukan peran dan tanggung jawab

Penanganan peristiwa keamanan membutuhkan disiplin lintas organisasi dan keinginan untuk bertindak. Dalam struktur organisasi Anda, harus ada banyak orang yang bertanggung jawab,

akuntabel, memberi konsultasi, atau terus mendapatkan informasi selama insiden, seperti perwakilan dari sumber daya manusia (SDM), tim eksekutif, dan legal. Pertimbangkan peran dan tanggung jawab ini, dan apakah pihak ketiga harus dilibatkan. Perhatikan bahwa banyak kawasan memiliki undang-undang setempat yang mengatur apa yang seharusnya dan tidak boleh dilakukan. Meskipun pembuatan bagan peran yang bertanggung jawab, akuntabel, memberi konsulatasi, mendapat informasi (RACI) untuk rencana respons keamanan terdengar birokratis, tindakan ini dapat memudahkan komunikasi yang cepat dan langsung serta menjelaskan secara gamblang kepemimpinan di berbagai tahap peristiwa.

Selama insiden, pelibatan pemilik dan developer aplikasi dan sumber daya yang terkena dampak adalah hal yang sangat penting karena mereka adalah pakar bidang (SME) yang dapat memberikan informasi dan konteks untuk membantu mengukur dampak. Pastikan untuk mempraktikkan dan membangun hubungan dengan developer dan pemilik aplikasi sebelum Anda mengandalkan keahlian mereka untuk merespons insiden. Pemilik aplikasi atau SME, seperti administrator atau rekayasawan cloud Anda, mungkin perlu bertindak dalam situasi ketika lingkungan kurang dikenal atau kompleks, atau ketika perespons tidak memiliki akses.

Terakhir, partner tepercaya dapat dilibatkan dalam penyelidikan atau respons karena mereka dapat memberikan keahlian tambahan dan pengawasan yang berharga. Ketika Anda tidak memiliki semua keterampilan ini di tim Anda sendiri, Anda mungkin ingin menggunakan bantuan dari pihak eksternal.

Pahami dukungan dan tim respons AWS

- AWS Support
  - AWS Support menawarkan berbagai paket yang menyediakan akses ke alat dan keahlian yang mendukung kesuksesan dan kesehatan operasional solusi AWS Anda. Jika Anda memerlukan dukungan teknis dan lebih banyak sumber daya untuk membantu merencanakan, men-deploy, dan mengoptimalkan lingkungan AWS Anda, Anda dapat memilih paket dukungan yang paling sesuai dengan kasus penggunaan AWS Anda.
  - Pertimbangkan Pusat Dukungan di AWS Management Console (diperlukan login) sebagai titik kontak utama untuk mendapatkan dukungan atas masalah yang memengaruhi sumber daya AWS Anda. Akses ke AWS Support dikontrol oleh AWS Identity and Access Management. Untuk informasi selengkapnya tentang mendapatkan akses ke fitur AWS Support, lihat Memulai dengan AWS Support.
- Tim Respons Insiden Pelanggan (CIRT) AWS

  - Tim Respons Insiden Pelanggan (CIRT) AWS adalah tim AWS global 24/7 khusus yang memberikan dukungan kepada pelanggan selama peristiwa keamanan aktif di sisi pelanggan dalam Model Tanggung Jawab Bersama AWS.
  - Ketika CIRT AWS mendukung Anda, mereka memberikan bantuan dengan evaluasi awal dan pemulihan untuk peristiwa keamanan aktif di AWS. Mereka dapat membantu menganalisis akar masalah melalui penggunaan log layanan AWS dan memberi Anda saran-saran pemulihan. Mereka juga dapat memberikan rekomendasi dan praktik terbaik keamanan untuk membantu Anda menghindari peristiwa keamanan di masa depan.
  - Pelanggan AWS dapat melibatkan CIRT AWS melalui kasus AWS Support.
- Dukungan respons DDoS
  - AWS menawarkan AWS Shield, yang menyediakan layanan perlindungan penolakan layanan terdistribusi (DDoS) yang terkelola yang melindungi aplikasi web yang berjalan di AWS. Shield menyediakan deteksi yang selalu aktif dan mitigasi inline otomatis yang dapat meminimalkan waktu henti dan latensi aplikasi, sehingga pelanggan tidak perlu melibatkan AWS Support untuk mendapatkan manfaat dari perlindungan DDoS. Terdapat dua tingkatan Shield: AWS Shield Standard dan AWS Shield Advanced. Untuk mempelajari perbedaan antara kedua tingkatan ini, lihat Dokumentasi fitur Shield.
- AWS Managed Services (AMS)
  - AWS Managed Services (AMS) menyediakan manajemen yang berkelanjutan untuk infrastruktur AWS Anda, sehingga Anda dapat berfokus pada aplikasi Anda. Dengan mengimplementasikan praktik terbaik untuk memelihara infrastruktur Anda, AMS membantu mengurangi overhead dan risiko operasional. AMS mengotomatiskan kegiatan umum seperti permintaan perubahan, pemantauan, manajemen patch, keamanan, dan layanan pencadangan serta menyediakan layanan siklus hidup penuh untuk menyediakan, menjalankan, dan mendukung infrastruktur Anda.
  - AMS bertanggung jawab untuk deployment serangkaian kontrol detektif keamanan dan menyediakan respons lini depan selama 24/7 terhadap peringatan. Saat peringatan dimulai, AMS mengikuti rangkaian standar playbook otomatis dan manual untuk memastikan konsistensi respons. Playbook ini dibagikan kepada pelanggan AMS selama orientasi sehingga mereka dapat mengembangkan dan mengoordinasikan respons dengan AMS.

Kembangkan rencana respons insiden

Rencana respons insiden dirancang untuk menjadi landasan bagi program dan strategi respons insiden Anda. Rencana respons insiden harus dalam dokumen resmi. Rencana respons insiden biasanya menyertakan bagian-bagian ini:

- Ikhtisar tim respons insiden: Menguraikan tujuan dan fungsi tim respons insiden.
- Peran dan tanggung jawab: Mencantumkan daftar pemangku kepentingan respons insiden dan memperinci peran mereka ketika insiden terjadi.
- Rencana komunikasi: Berisi detail informasi kontak dan cara Anda berkomunikasi selama insiden.
- Metode komunikasi cadangan: Salah satu praktik terbaik adalah memiliki komunikasi di luar saluran normal (out-of-band) sebagai cadangan untuk komunikasi insiden. Contoh aplikasi yang menyediakan saluran komunikasi out-of-band yang aman adalah AWS Wickr.
- Tahapan respons insiden dan tindakan yang harus dilakukan: Menyebutkan satu per satu tahapan respons insiden (misalnya mendeteksi, menganalisis, memberantas, mengendalikan, dan memulihkan), termasuk tindakan umum yang harus dilakukan dalam tahapan-tahapan tersebut.
- Penetapan tingkat keparahan dan prioritas insiden: Menjelaskan cara mengklasifikasikan tingkat keparahan insiden, cara memprioritaskan insiden, dan bagaimana penetapan keparahan memengaruhi prosedur eskalasi.

Meskipun bagian-bagian ini sudah umum ada di perusahaan dari berbagai ukuran dan industri, rencana respons insiden di setiap organisasi berbeda-beda. Anda perlu membangun rencana respons insiden yang paling cocok untuk organisasi Anda.

Sumber daya

Praktik terbaik terkait:

- SEC04 (Bagaimana cara mendeteksi dan menyelidiki peristiwa keamanan?)

Dokumen terkait:

- Respons Insiden Keamanan AWS
- NIST: Panduan Penanganan Insiden Keamanan Komputer

SEC10-BP03 Menyiapkan kemampuan forensik

Sebelum terjadinya insiden keamanan, pertimbangkan untuk mengembangkan kemampuan forensik untuk mendukung investigasi event keamanan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Konsep dari forensik on-premise tradisional berlaku untuk AWS. Untuk informasi penting untuk mulai membangun kemampuan forensik di AWS Cloud, lihat Strategi lingkungan investigasi forensik di AWS Cloud.

Setelah Anda menyiapkan lingkungan dan struktur Akun AWS Anda untuk forensik, tentukan teknologi yang diperlukan untuk secara efektif melakukan metodologi yang baik secara forensik di empat fase:

- Pengumpulan: Kumpulkan log AWS yang relevan, seperti AWS Config, Log Aliran VPC, dan log tingkat host. Kumpulkan snapshot, cadangan, dan dump memori dari sumber daya AWS yang terkena dampak jika tersedia.
- Pemeriksaan: Periksa data yang dikumpulkan dengan mengekstraksi dan menilai informasi yang relevan.
- Analisis: Lakukan analisis data yang dikumpulkan untuk memahami insiden dan menarik kesimpulan darinya.
- Pelaporan: Sajikan informasi yang dihasilkan dari fase analisis.

Langkah implementasi

Persiapkan lingkungan forensik Anda

AWS Organizations membantu Anda mengelola dan mengatur lingkungan AWS secara terpusat saat Anda mengembangkan dan menskalakan sumber daya AWS. Sebuah organisasi AWS menggabungkan Akun AWS Anda sehingga Anda dapat mengelolanya sebagai satu unit. Anda dapat menggunakan unit organisasi (OU) untuk mengumpulkan akun-akun untuk dikelola sebagai satu unit.

Untuk respons insiden, ada manfaatnya memiliki struktur Akun AWS yang mendukung fungsi respons insiden, yang mencakup OU Keamanan dan OU Forensik. Di dalam OU keamanan, Anda harus memiliki akun untuk:

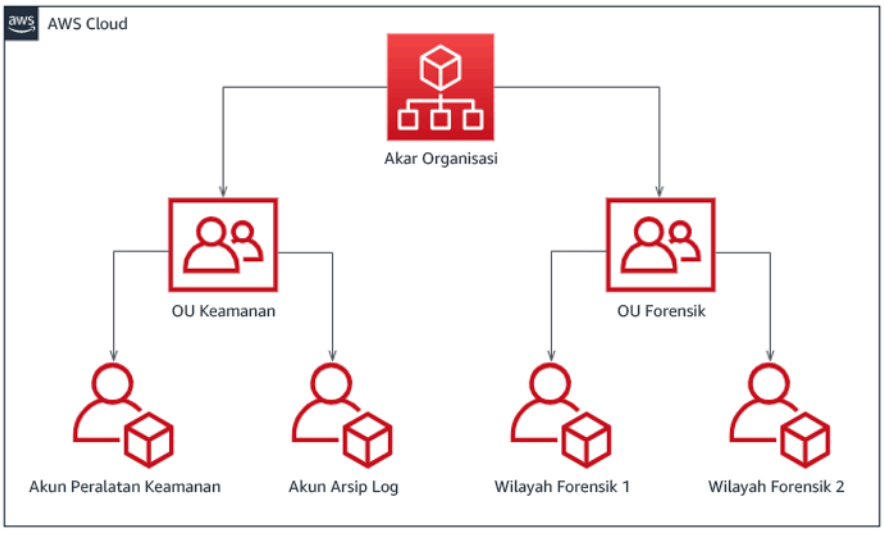
- Pengarsipan log: Agregasikan log dalam Akun AWS pengarsipan log dengan izin terbatas.

- Alat keamanan: Pusatkan layanan keamanan di Akun AWS alat keamanan. Akun ini beroperasi sebagai administrator yang didelegasikan untuk layanan keamanan.

Dalam OU forensik, Anda memiliki opsi untuk mengimplemetasikan satu atau beberapa akun forensik untuk setiap Region tempat Anda beroperasi, tergantung mana yang paling sesuai untuk model bisnis dan operasional Anda. Jika Anda membuat akun forensik per Region, Anda dapat memblokir pembuatan sumber daya AWS di luar Region tersebut dan mengurangi risiko sumber daya disalin ke region yang tidak diinginkan. Misalnya, jika Anda hanya beroperasi di US East (N. Virginia) Region (`us-east-1`) dan US West (Oregon) (`us-west-2`), maka Anda akan memiliki dua akun di OU forensik: satu untuk `us-east-1` dan satu untuk `us-west-2`.

Anda dapat membuat Akun AWS forensik untuk beberapa Region. Anda harus berhati-hati dalam menyalin sumber daya AWS ke akun tersebut untuk memastikan keselarasan dengan persyaratan kedaulatan data Anda. Karena diperlukan waktu untuk menyediakan akun baru, akun forensik harus dibuat dan dilengkapi jauh sebelum insiden sehingga tim perespons dapat dipersiapkan untuk menggunakannya secara efektif untuk merespons.

Diagram berikut ini menampilkan contoh struktur akun termasuk OU forensik dengan akun forensik per Region:



Struktur akun per Region untuk respons insiden

Rekam cadangan dan snapshot

Menyiapkan cadangan sistem dan basis data utama sangat penting untuk pemulihan dari insiden keamanan dan untuk tujuan forensik. Dengan cadangan di tempat, Anda dapat memulihkan sistem

Anda ke keadaan aman sebelumnya. Pada AWS, Anda dapat membuat snapshot berbagai sumber daya. Snapshot memberi Anda cadangan titik waktu dari sumber daya tersebut. Ada banyak layanan AWS yang dapat mendukung Anda dalam pencadangan dan pemulihan. Untuk detail tentang layanan dan pendekatan ini untuk pencadangan dan pemulihan, lihat Panduan Preskriptif Pencadangan dan Pemulihan dan Gunakan cadangan untuk memulihkan dari insiden keamanan.

Terutama berkaitan dengan situasi seperti ransomware, cadangan Anda harus terlindungi dengan baik. Untuk panduan tentang mengamankan cadangan Anda, lihat 10 praktik terbaik keamanan untuk mengamankan cadangan di AWS. Selain mengamankan cadangan Anda, Anda harus secara rutin menguji proses pencadangan dan pemulihan Anda untuk memastikan teknologi dan proses yang Anda miliki berfungsi seperti yang diharapkan.

Mengotomatiskan forensik

Selama event keamanan, tim respons insiden Anda harus dapat mengumpulkan dan menganalisis bukti dengan cepat sambil mempertahankan akurasi selama jangka waktu sebelum dan sesudah event (seperti menangkap log yang berkaitan dengan event atau sumber daya tertentu atau mengumpulkan dump memori suatu instans Amazon EC2). Ini adalah hal yang menantang dan memakan waktu bagi tim respons insiden untuk mengumpulkan bukti yang relevan secara manual, terutama di sejumlah besar instans dan akun. Selain itu, pengumpulan manual bisa rentan terhadap kesalahan manusia. Untuk alasan-alasan ini, Anda harus mengembangkan dan mengimplementasikan otomatisasi untuk forensik sebanyak mungkin.

AWS menawarkan sejumlah sumber daya otomatisasi untuk forensik, yang tercantum di bagian Sumber Daya berikut ini. Sumber daya ini adalah contoh-contoh pola forensik yang telah kami kembangkan dan telah diimplementasikan oleh pelanggan. Meskipun merupakan arsitektur referensi yang berguna untuk memulai, pertimbangkan untuk memodifikasinya atau membuat pola otomatisasi forensik baru berdasarkan lingkungan, persyaratan, alat, dan proses forensik Anda.

Sumber daya

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS - Mengembangkan Kemampuan Forensik
- Panduan Respons Insiden Keamanan AWS - Sumber Daya Forensik
- Strategi lingkungan investigasi forensik di AWS Cloud
- Cara mengotomatiskan pengumpulan disk forensik di AWS
- Panduan Preskriptif AWS - Mengotomatiskan respons insiden dan forensik

Video terkait:

- Mengotomatiskan Respons Insiden dan Forensik

Contoh terkait:

- Respons Insiden Otomatis dan Kerangka Forensik
- Orkestrator Forensik Otomatis untuk Amazon EC2

SEC10-BP04 Mengembangkan dan menguji playbook respons insiden keamanan

Bagian penting dari penyiapan proses respons insiden Anda adalah mengembangkan playbook. Playbook respons insiden memberikan serangkaian panduan preskriptif dan langkah-langkah yang harus diikuti ketika event keamanan terjadi. Memiliki struktur dan langkah yang jelas menyederhanakan respons dan mengurangi kemungkinan kesalahan manusia.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Playbook harus dibuat untuk skenario insiden seperti:

- Insiden yang diperkirakan: Playbook harus dibuat untuk insiden yang Anda antisipasi. Ini mencakup ancaman seperti denial of service (DoS), ransomware, dan kompromi kredensial.
- Temuan atau pemberitahuan keamanan yang diketahui: Playbook harus dibuat untuk temuan dan pemberitahuan keamanan Anda yang diketahui, seperti temuan . Anda mungkin menerima temuan GuardDuty dan berpikir, “Lalu apa?” Untuk mencegah kesalahan penanganan atau pengabaian temuan GuardDuty, buat playbook untuk setiap temuan GuardDuty potensial. Beberapa detail dan panduan perbaikan dapat ditemukan di dokumentasi GuardDuty. Perlu dicatat bahwa GuardDuty tidak diaktifkan secara default dan dikenakan biaya. Untuk detail selengkapnya tentang GuardDuty, lihat Lampiran A: Definisi kemampuan cloud - Visibilitas dan pembuatan pemberitahuan.

Playbook harus berisi langkah-langkah teknis untuk diselesaikan oleh analis keamanan untuk menyelidiki dan merespons insiden keamanan potensial secara memadai.

Langkah implementasi

Item yang perlu disertakan dalam playbook meliputi:

- Gambaran umum playbook: Risiko atau skenario insiden apa yang ditangani oleh playbook ini? Apa tujuan dari playbook?
- Prasyarat: Log, mekanisme deteksi, dan alat otomatis apa yang diperlukan untuk skenario insiden ini? Apa notifikasi yang diharapkan?
- Komunikasi dan informasi eskalasi: Siapa yang terlibat dan apa informasi kontak mereka? Apa tanggung jawab setiap pemangku kepentingan?
- Langkah-langkah respons: Di seluruh fase respons insiden, langkah taktis apa yang harus dilakukan? Kueri apa yang harus dijalankan analis? Kode apa yang harus dijalankan untuk mencapai hasil yang diinginkan?
  - Deteksi: Bagaimana insiden akan dideteksi?
  - Analisis: Bagaimana cakupan dampak ditentukan?
  - Membendung: Bagaimana insiden akan diisolasi untuk membatasi cakupan?
  - Memberantas: Bagaimana ancaman akan disingkirkan dari lingkungan?
  - Memulihkan: Bagaimana sistem atau sumber daya yang terdampak akan dikembalikan ke produksi?
- Hasil yang diharapkan: Setelah kueri dan kode dijalankan, apa hasil yang diharapkan dari playbook?

Sumber daya

Praktik terbaik Well-Architected terkait:

- SEC10-BP02 - Membuat rencana manajemen insiden

Dokumen terkait:

- Kerangka Kerja untuk Playbook Respons Insiden
- Mengembangkan Playbook Respons Insiden Anda sendiri
- Contoh Playbook Respons Insiden
- Membangun runbook respons insiden AWS menggunakan playbook Jupyter dan CloudTrail Lake

SEC10-BP05 Menyediakan akses di awal

Verifikasi staf respons insiden memiliki akses yang benar yang telah disediakan sebelumnya di AWS untuk mengurangi waktu yang diperlukan untuk penyelidikan hingga pemulihan.

Antipola umum:

- Menggunakan akun root untuk merespons insiden.
- Mengubah akun-akun pengguna yang ada.
- Memanipulasi izin IAM secara langsung saat menyediakan peningkatan hak akses yang sedang dibutuhkan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

AWS menyarankan Anda untuk sebisa mungkin mengurangi atau menghilangkan kebergantungan pada kredensial berumur panjang, dan memilih kredensial sementara dan mekanisme peningkatan hak akses hanya saat diperlukan. Kredensial berumur panjang rentang terkena risiko keamanan dan meningkatkan biaya overhead operasional. Untuk sebagian besar tugas manajemen, serta tugas respons insiden, kami menyarankan Anda untuk mengimplementasikan federasi identitas bersamaan dengan peningkatan sementara untuk akses administratif. Di model ini, seorang pengguna meminta peningkatan ke tingkat hak akses yang lebih tinggi (seperti peran respons insiden) dan, apabila pengguna tersebut memenuhi syarat peningkatan hak, permintaan tersebut dikirimkan ke seorang pemberi persetujuan. Jika permintaan tersebut disetujui, pengguna menerima serangkaian kredensial AWS sementara yang dapat digunakan untuk menyelesaikan tugas-tugas mereka. Setelah kredensial ini kedaluwarsa, pengguna harus mengirimkan permintaan peningkatan baru.

Kami menyarankan penggunaan peningkatan hak akses sementara di sebagian besar skenario respons insiden. Cara tepat untuk melakukannya adalah dengan menggunakan AWS Security Token Service dan kebijakan sesi untuk membuat cakupan akses.

Terdapat skenario di mana identitas terfederasi tidak tersedia, seperti:

- Pemadaman yang berkaitan dengan penyedia identitas (IdP) yang terganggu.
- Kesalahan konfigurasi atau kesalahan manusiawi yang menyebabkan rusaknya sistem manajemen akses terfederasi.

- Aktivitas berbahaya seperti peristiwa distributed denial of service (DDoS) atau yang menyebabkan sistem tidak tersedia.

Pada kasus-kasus di atas, harus terdapat akses mendesak (break glass) yang dikonfigurasi untuk mengizinkan penyelidikan dan perbaikan peristiwa secara cepat. Sebaiknya gunakan juga pengguna IAM dengan izin yang tepat untuk menjalankan tugas dan mengakses sumber daya AWS. Gunakan kredensial root hanya untuk tugas yang memerlukan akses pengguna root. Untuk memverifikasi bahwa tim respons insiden memiliki tingkat akses yang tepat ke AWS dan sistem yang relevan lainnya, sebaiknya sediakan akun-akun pengguna khusus sejak awal. Akun pengguna tersebut memerlukan akses istimewa, dan harus dikontrol dan dipantau secara ketat. Akun-akun tersebut harus dibuat dengan hak akses paling sedikit yang diperlukan untuk menjalankan tugas yang diperlukan, dan tingkat akses harus didasarkan pada playbook yang dibuat sebagai bagian dari rencana manajemen insiden.

Gunakan pengguna dan peran yang dibuat khusus sebagai praktik terbaik. Peningkatan akses pengguna atau peran sementara melalui penambahan kebijakan IAM menjadikannya tidak jelas terkait akses apa yang dimiliki pengguna selama insiden, dan terdapat risiko tidak dicabutnya peningkatan hak akses tersebut.

Penting untuk menghapus dependensi sebanyak mungkin untuk memastikan akses dapat diperoleh dalam sebanyak mungkin skenario kegagalan. Untuk mendukung hal ini, buatlah playbook untuk memastikan pengguna respons insiden dibuat sebagai pengguna AWS Identity and Access Management di dalam akun keamanan khusus, dan bukan dikelola melalui solusi Federasi atau masuk tunggal (SSO) yang ada. Tiap-tiap perespons harus memiliki akun dengan nama mereka sendiri. Konfigurasi akun harus menegakkan kebijakan kata sandi yang kuat dan autentikasi multi-faktor (MFA). Jika playbook respons insiden hanya memerlukan akses ke AWS Management Console, pengguna tidak boleh memiliki kunci akses yang dikonfigurasi dan harus dilarang secara tegas untuk membuat kunci akses. Hal ini dapat dikonfigurasi dengan kebijakan IAM atau kebijakan kontrol layanan (SCP) sebagaimana disebutkan dalam Praktik Terbaik Keamanan AWS untu AWS Organizations SCP. Pengguna tidak boleh memiliki hak ases selain kemampuan untuk mengambil peran respons insiden di akun-akun lainnya.

Selama insiden, mungkin diperlukan pemberian akses ke individu internal atau eksternal untuk mendukung aktivitas penyelidikan, perbaikan, atau pemulihan. Pada kasus ini, gunakan mekanisme playbook yang disebutkan sebelumnya, dan harus ada proses untuk memverifikasi bahwa akses tambahan apa pun segera dicabut setelah insiden selesai.

Untuk memastikan bahwa penggunaan peran respons insiden dapat dipantau dan diaudit dengan layak, penting untuk tidak membagikan akun pengguna IAM yang dibuat untuk tujuan ini kepada individu lain, serta tidak menggunakan pengguna root Akun AWS kecuali diperlukan untuk tugas tertentu. Jika pengguna root diperlukan (sebagai contoh, akses IAM ke akun tertentu tidak tersedia), gunakan proses terpisah dengan playbook yang tersedia untuk memverifikasi ketersediaan kata sandi pengguna root dan token MFA.

Untuk mengonfigurasi kebijakan IAM untuk peran respons insiden, pertimbangkan menggunakan IAM Access Analyzer untuk menghasilkan kebijakan berdasarkan log AWS CloudTrail. Untuk melakukannya, berikan akses administrator ke peran respons insiden di akun non-produksi dan jalankan playbook Anda. Setelah selesai, kebijakan dapat dibuat yang hanya mengizinkan tindakan yang diambil. Kebijakan ini kemudian dapat diterapkan ke semua peran respons insiden di semua akun. Anda mungkin ingin membuat kebijakan IAM terpisah untuk setiap playbook untuk mempermudah manajemen dan audit. Contoh playbook dapat mencakup rencana respons untuk ransomware, pembobolan data, hilangnya akses produksi, dan skenario lain.

Gunakan akun pengguna respons insiden untuk mengambil peran IAM respons insiden khusus di Akun AWS lain. Peran-peran ini harus dikonfigurasi hanya agar dapat diambil oleh pengguna di akun keamanan, dan hubungan kepercayaan harus mewajibkan bahwa principal pemanggil telah mengautentikasi menggunakan MFA. Peran-peran tersebut harus menggunakan kebijakan IAM dengan cakupan yang ketat untuk mengontrol akses. Pastikan bahwa semua permintaan `AssumeRole` untuk peran-peran ini dicatat dalam log di CloudTrail dan dibuatkan peringatan, dan bahwa tindakan apa pun yang diambil menggunakan peran-peran ini dicatat dalam log.

Sangat disarankan akun pengguna IAM serta IAM role disebutkan secara jelas agar dapat ditemukan dengan mudah di log CloudTrail. Contohnya adalah dengan menamai akun IAM dengan *<USER_ID>*`-BREAK-GLASS` dan IAM role dengan `BREAK-GLASS-ROLE`.

CloudTrail digunakan untuk membuat log aktivitas API di akun AWS Anda dan harus digunakan untuk mengonfigurasi peringatan penggunaan peran respons insiden. Lihat postingan blog tentang konfigurasi perintanan saat kunci root digunakan. Instruksi dapat dimodifikasi untuk mengonfigurasi filter-ke-filter metrik Amazon CloudWatch pada peristiwa `AssumeRole` terkait dengan IAM role respons insiden:

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !
= "AwsServiceEvent" }
```

Karena peran respons insiden kemungkinan memiliki tingkat akses yang tinggi, peringatan-peringatan ini harus menjangkau grup yang luas dan ditindaklanjuti segera.

Selama insiden, terdapat kemungkinan bahwa perespons mungkin memerlukan akses ke sistem yang tidak diamankan secara langsung oleh IAM. Sistem-sistem tersebut dapat mencakup instans Amazon Elastic Compute Cloud, basis data Amazon Relational Database Service, atau platform perangkat lunak sebagai layanan (SaaS). Sangat disarankan untuk tidak menggunakan protokol native seperti SSH atau RDP, melainkan AWS Systems Manager Session Manager untuk semua akses administratif ke instans Amazon EC2. Akses ini dapat dikontrol menggunakan IAM, yang aman dan diaudit. Memungkinkan juga untuk mengotomatisasi bagian-bagian playbook Anda menggunakan dokumen AWS Systems Manager Run Command, yang dapat mengurangi kesalahan pengguna dan mempercepat waktu pemulihan. Untuk akses ke basis data dan alat-alat pihak ketiga, kami sarankan menyimpan kredensial akses di AWS Secrets Manager dan memberikan akses ke peran perespons insiden.

Terakhir, manajemen akun pengguna IAM perespons insiden harus ditambahkan ke proses Joiners, Movers, dan Leavers Anda dan ditinjau serta diuji secara berkala untuk memastikan bahwa yang diizinkan hanyalah akses yang diinginkan.

Sumber daya

Dokumen terkait:

- Mengelola peningkatan akses sementara ke lingkungan AWS Anda
- Panduan Respons Insiden Keamanan AWS
- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- Mengatur kebijakan kata sandi akun untuk pengguna IAM
- Menggunakan autentikasi multi-faktor (MFA) di AWS
- Mengonfigurasi Akses Lintas Akun dengan MFA
- Menggunakan IAM Access Analyzer untuk menghasilkan kebijakan IAM
- Praktik Terbaik untuk Kebijakan Kontrol Layanan AWS Organizations di Lingkungan Multi-akun
- Cara Menerima Notifikasi Ketika Kunci Akses Root Akun AWS Anda Digunakan
- Membuat izin sesi mendetail menggunakan kebijakan terkelola IAM

Video terkait:

- Mengotomatiskan Respons Insiden dan Forensik di AWS
- Panduan mandiri untuk runbook, laporan insiden, dan respons insiden
- Bersiap dan merespons insiden keamanan di lingkungan AWS Anda

Contoh terkait:

- Lab: Penyiapan dan Pengguna Root Akun AWS
- Lab: Respons insiden dengan Konsol AWS dan CLI

SEC10-BP06 Melakukan deployment alat di awal

Pastikan personel keamanan sejak awal telah melakukan deployment alat yang tepat untuk mengurangi waktu investigasi melalui pemulihan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Untuk mengotomatiskan respons keamanan dan fungsi operasi, Anda dapat menggunakan set API dan alat yang komprehensif dari AWS. Anda dapat sepenuhnya mengotomatiskan manajemen identitas, keamanan jaringan, perlindungan data, dan kemampuan pemantauan, serta menyediakannya menggunakan metode pengembangan perangkat lunak populer yang sudah Anda gunakan. Saat Anda membangun otomatisasi keamanan, sistem Anda dapat memantau, meninjau, dan menginisiasi respons, tanpa memerlukan orang untuk memantau posisi keamanan Anda dan memberikan reaksi terhadap peristiwa secara manual.

Jika tim respons insiden Anda terus merespons peringatan dengan cara yang sama, mereka berisiko mengalami kelelahan alarm (alarm fatigue). Seiring berjalannya waktu, tim dapat menjadi tidak peka terhadap peringatan sehingga dapat membuat kesalahan saat menangani situasi biasa atau melewatkan peringatan yang tidak biasa. Otomatisasi membantu mencegah kelelahan alarm dengan menggunakan fungsi yang memproses peringatan biasa dan repetitif, sehingga manusia cukup menangani insiden yang sensitif dan unik. Integrasi sistem deteksi anomali, seperti Amazon GuardDuty, Wawasan AWS CloudTrail, dan Deteksi Anomali Amazon CloudWatch, dapat mengurangi beban dari peringatan umum berbasis ambang batas.

Anda dapat memperbaiki proses manual dengan mengotomatiskan langkah-langkah dalam proses secara terprogram. Setelah Anda menentukan perbaikan pola pada peristiwa, Anda dapat

menguraikan pola tersebut menjadi logika yang dapat ditindaklanjuti, dan menulis kode untuk menjalankan logika tersebut. Pemberi respons selanjutnya dapat menjalankan kode tersebut untuk memperbaiki masalah. Seiring berjalannya waktu, Anda dapat mengotomatiskan lebih banyak langkah, dan pada akhirnya secara otomatis menangani semua jenis insiden yang biasa muncul.

Selama penyelidikan keamanan, Anda harus dapat meninjau log yang relevan untuk mencatat dan memahami seluruh cakupan serta garis waktu insiden. Log juga diperlukan untuk pembuatan peringatan yang mengindikasikan bahwa tindakan tertentu telah terjadi. Sangat penting untuk memilih, mengaktifkan, menyimpan, dan menyiapkan mekanisme kueri dan pengambilan, serta pembuatan peringatan. Selain itu, cara yang efektif untuk menyediakan alat untuk mencari data log adalah Amazon Detective.

AWS menawarkan lebih dari 200 layanan cloud dan ribuan fitur. Kami menyarankan Anda meninjau layanan yang dapat mendukung dan menyederhanakan strategi respons insiden Anda.

Selain pencatatan log, Anda harus mengembangkan dan menerapkan strategi pemberian tag. Pemberian tag dapat membantu memberikan konteks seputar tujuan sebuah sumber daya AWS. Pemberian tag juga dapat digunakan untuk otomatisasi.

Langkah implementasi

Pilih dan atur log untuk analisis dan pembuatan peringatan

Lihat dokumentasi berikut tentang cara mengonfigurasi pencatatan log untuk respons insiden:

- Strategi pencatatan log untuk respons insiden keamanan
- SEC04-BP01 Mengonfigurasi pencatatan log layanan dan aplikasi

Aktifkan layanan keamanan untuk mendukung deteksi dan respons

AWS menyediakan kemampuan deteksi, pencegahan, dan responsif asli, dan layanan lainnya dapat digunakan untuk merancang solusi keamanan khusus. Untuk daftar layanan yang paling relevan untuk respons insiden keamanan, lihat Definisi kemampuan cloud.

Mengembangkan dan menerapkan strategi pemberian tag

Memperoleh informasi kontekstual tentang kasus penggunaan bisnis dan pemangku kepentingan internal yang relevan seputar sumber daya AWS bisa jadi sulit dilakukan. Salah satu cara untuk melakukannya adalah dalam bentuk tag, yang menetapkan metadata ke sumber daya AWS

Anda dan terdiri dari kunci dan nilai yang ditentukan pengguna. Anda dapat membuat tag untuk mengategorikan sumber daya berdasarkan tujuan, pemilik, lingkungan, jenis data yang diproses, dan kriteria lain pilihan Anda.

Memiliki strategi pemberian tag yang konsisten dapat mempercepat waktu respons dan meminimalkan waktu yang dihabiskan untuk konteks organisasi dengan memungkinkan Anda mengidentifikasi dan membedakan informasi kontekstual tentang sumber daya AWS dengan cepat. Tag juga dapat berfungsi sebagai mekanisme untuk memulai otomatisasi respons. Untuk detail selengkapnya tentang apa yang harus diberi tag, lihat Memberi tag pada sumber daya AWS Anda. Anda harus terlebih dahulu menentukan tag yang ingin Anda terapkan di seluruh organisasi Anda. Setelah itu, Anda akan menerapkan dan menegakkan strategi pemberian tag ini. Untuk detail selengkapnya tentang implementasi dan penegakan, lihat Menerapkan strategi pemberian tag sumber daya AWS menggunakan Kebijakan Tag dan Kebijakan Kontrol Layanan (SCP) AWS.

Sumber daya

Praktik terbaik Well-Architected terkait:

- SEC04-BP01 Mengonfigurasi pencatatan log layanan dan aplikasi
- SEC04-BP02 Menganalisis log, temuan, dan metrik secara terpusat

Dokumen terkait:

- Strategi pencatatan log untuk respons insiden keamanan
- Definisi kemampuan cloud respons insiden

Contoh terkait:

- Deteksi dan Respons Ancaman dengan Amazon GuardDuty dan Amazon Detective
- Lokakarya Security Hub
- Manajemen Kerentanan dengan Amazon Inspector

SEC10-BP07 Menjalankan simulasi

Organisasi tumbuh dan berkembang dari waktu ke waktu, begitu juga dengan lanskap ancaman. Oleh karena itu, penting untuk terus-menerus mengkaji kemampuan Anda dalam merespons insiden. Menjalankan simulasi (juga dikenal dengan nama game day) adalah salah satu metode yang dapat

digunakan untuk melakukan penilaian ini. Simulasi menggunakan skenario peristiwa keamanan dunia nyata yang dirancang untuk meniru taktik, teknik, dan prosedur (TTP) pelaku ancaman dan memungkinkan organisasi untuk melatih dan mengevaluasi kemampuan respons insiden mereka dengan merespons peristiwa dunia maya tiruan ini, yang bisa saja benar-benar terjadi di dunia nyata.

Manfaat menerapkan praktik terbaik ini: Simulasi memiliki berbagai manfaat:

- Memvalidasi kesiapan dunia maya dan membangun kepercayaan diri perespons insiden Anda.
- Menguji akurasi dan efisiensi alat dan alur kerja.
- Menyempurnakan metode komunikasi dan eskalasi yang selaras dengan rencana respons insiden Anda.
- Memberikan kesempatan untuk merespons vektor-vektor yang kurang umum.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Terdapat tiga jenis simulasi utama:

- Latihan simulasi meja: Pendekatan simulasi meja adalah sesi berbasis diskusi yang melibatkan berbagai pemangku kepentingan respons insiden untuk melatih peran dan tanggung jawab menggunakan alat komunikasi dan playbook yang lazim. Fasilitasi latihan umumnya dapat dilakukan selama sehari penuh di lokasi virtual, lokasi fisik, atau gabungan keduanya. Karena berbasis diskusi, latihan meja berfokus pada proses, orang, dan kolaborasi. Teknologi merupakan bagian tak terpisahkan dari diskusi, tetapi penggunaan nyata alat atau skrip respons insiden umumnya bukan bagian dari latihan meja.
- Latihan tim ungu: Latihan tim ungu meningkatkan level kolaborasi antara perespons insiden (tim biru) dan yang bermain sebagai pelaku ancaman (tim merah). Tim biru terdiri dari anggota pusat operasi keamanan (SOC), tetapi juga bisa melibatkan pemangku kepentingan lain yang akan terlibat selama peristiwa dunia maya nyata. Tim merah terdiri dari tim uji penetrasi atau pemangku kepentingan utama yang terlatih dalam hal keamanan ofensif. Tim merah bekerja secara kolaboratif dengan fasilitator latihan saat merancang skenario sehingga skenario benar-benar akurat dan layak. Selama latihan tim ungu, fokus utamanya adalah mekanisme deteksi, alat, dan prosedur operasi standar (SOP) yang mendukung upaya respons insiden.
- Latihan tim merah: Selama latihan tim merah, pelanggaran (tim merah) melakukan simulasi untuk mencapai satu tujuan atau serangkaian tujuan tertentu dari ruang lingkup yang telah ditentukan. Para pelindung (tim biru) tidak harus mengetahui ruang lingkup dan durasi latihan, sehingga

penilaian menjadi lebih realistis tentang cara mereka merespons insiden yang sebenarnya. Karena latihan tim merah bisa bersifat invasif, berhati-hatilah dan terapkan kontrol untuk memverifikasi bahwa latihan tidak menyebabkan kerusakan nyata pada lingkungan Anda.

Pertimbangkan untuk memfasilitasi simulasi dunia maya secara berkala. Setiap jenis latihan dapat memberikan manfaat unik bagi peserta dan organisasi secara keseluruhan, sehingga Anda dapat memilih untuk memulai dengan jenis simulasi yang lebih sederhana (seperti latihan meja) dan secara bertahap beralih ke jenis simulasi yang lebih kompleks (latihan tim merah). Anda harus memilih jenis simulasi berdasarkan kematangan keamanan dan sumber daya Anda, serta hasil yang Anda inginkan. Beberapa pelanggan mungkin tidak mau melakukan latihan tim merah disebabkan kompleksitas dan biayanya.

Langkah implementasi

Terlepas dari jenis simulasi yang Anda pilih, simulasi umumnya mengikuti langkah-langkah implementasi berikut:

1. Tentukan elemen latihan inti: Tentukan skenario simulasi dan tujuan simulasi. Keduanya harus atas izin pimpinan.
2. Identifikasi pemangku kepentingan utama: Minimal, latihan membutuhkan fasilitator dan peserta latihan. Tergantung skenario, pemangku kepentingan tambahan seperti pimpinan tim legal, komunikasi, atau eksekutif dapat dilibatkan.
3. Bangun dan uji skenario: Skenario mungkin perlu dirombak selama penyusunan apabila elemen tertentu tidak layak. Skenario akhir adalah output yang diharapkan pada tahap ini.
4. Fasilitasi simulasi: Jenis simulasi menentukan fasilitas yang digunakan (skenario berbasis kertas versus skenario tersimulasi yang sangat teknis). Fasilitator harus menyelaraskan taktik fasilitas mereka dengan objek latihan dan mereka harus sebisa mungkin melibatkan semua peserta latihan untuk memberikan manfaat paling optimal.
5. Kembangkan laporan pascatindakan (AAR): Identifikasi area yang berjalan dengan baik, area yang memerlukan perbaikan, dan potensi celah. AAR harus mengukur efektivitas simulasi serta respons tim terhadap simulasi peristiwa sehingga kemajuan dapat dilacak dari waktu ke waktu dengan simulasi di masa mendatang.

Sumber daya

Dokumen terkait:

- Panduan Respons Insiden AWS

Video terkait:

- AWS GameDay - Edisi Keamanan

Operasi

Operasi adalah inti dari pelaksanaan respons insiden. Di sinilah tindakan respons dan remediasi insiden keamanan terjadi. Operasi meliputi lima fase berikut: deteksi, analisis, pengendalian, pemberantasan, dan pemulihan. Deskripsi fase-fase ini serta tujuannya dapat ditemukan di dalam tabel berikut.

Fase	Tujuan
Deteksi	Mengidentifikasi potensi peristiwa keamanan.
Analisis	Menentukan apakah peristiwa keamanan merupakan sebuah insiden dan menilai cakupan insiden tersebut.
Pengendalian	Minimalkan dan batasi cakupan peristiwa keamanan.
Pemberantasan	Menghapus sumber daya atau artefak yang tidak sah yang berkaitan dengan peristiwa keamanan. Mengimplementasikan mitigasi yang menyebabkan insiden keamanan.
Pemulihan	Memulihkan sistem ke keadaan yang diketahui aman dan memantau sistem-sistem ini untuk memastikan ancaman tidak kembali.

Fase-fase ini harus berfungsi sebagai panduan ketika Anda merespons dan beroperasi terhadap insiden keamanan untuk merespons dengan cara yang efektif dan kuat. Tindakan aktual yang Anda ambil akan berbeda-beda tergantung insidennya. Insiden yang melibatkan ransomware, misalnya, akan memiliki serangkaian langkah respons yang berbeda-beda untuk diikuti daripada insiden yang

melibatkan bucket Amazon S3 publik. Selain itu, fase-fase ini tidak selalu terjadi secara berurutan. Setelah pengendalian dan pemberantasan, Anda mungkin perlu kembali ke analisis untuk memahami apakah tindakan Anda efektif.

Persiapan yang menyeluruh untuk personel, proses, dan teknologi Anda adalah kunci untuk efektivitas dalam operasi. Dengan demikian, ikuti praktik terbaik dari bagian Persiapan untuk dapat merespons peristiwa keamanan aktif secara efektif.

Untuk mempelajari lebih lanjut, lihat bagian Operasi Panduan Respons Insiden Keamanan AWS.

Aktivitas Pascainsiden

Lanskap ancaman senantiasa berubah dan kemampuan organisasi Anda juga harus sama-sama dinamis untuk melindungi lingkungan Anda secara efektif. Kunci untuk perbaikan yang berkelanjutan adalah mengiterasi hasil dari insiden dan simulasi Anda untuk meningkatkan kemampuan Anda untuk secara efektif mendeteksi, merespons, dan menyelidiki kemungkinan insiden keamanan, mengurangi kemungkinan kerentanan Anda, waktu untuk merespons, dan kembali ke operasi yang aman. Mekanisme berikut ini dapat membantu Anda memastikan organisasi Anda tetap siap dengan kemampuan dan pengetahuan terbaru untuk merespons secara efektif, apa pun situasinya.

Praktik terbaik

- SEC10-BP08 Menetapkan kerangka kerja untuk belajar dari insiden

SEC10-BP08 Menetapkan kerangka kerja untuk belajar dari insiden

Menerapkan kerangka kerja belajar dari pengalaman dan kemampuan analisis akar masalah tidak hanya dapat membantu meningkatkan kemampuan respons insiden, tetapi juga membantu mencegah insiden berulang. Dengan belajar dari setiap kejadian, Anda dapat membantu menghindari mengulangi kesalahan, paparan, atau kesalahan konfigurasi yang sama, sehingga tidak hanya meningkatkan postur keamanan Anda, tetapi juga meminimalkan waktu yang hilang untuk situasi yang dapat dicegah.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Penting untuk menerapkan kerangka kerja belajar dari pengalaman yang secara umum menetapkan dan mencapai poin-poin berikut ini:

- Kapan belajar dari pengalaman diadakan?
- Apa yang terlibat dalam proses belajar dari pengalaman?
- Bagaimana belajar dari pengalaman dilaksanakan?
- Siapa yang terlibat dalam prosesnya, dan bagaimana?
- Bagaimana area perbaikan akan diidentifikasi?
- Bagaimana Anda akan memastikan bawa perbaikan dilacak dan diimplementasikan secara efektif?

Kerangka kerja ini tidak boleh fokus pada individu atau menyalahkan individu, tetapi harus fokus pada perbaikan alat dan proses.

Langkah implementasi

Selain hasil tingkat tinggi yang dicantumkan sebelumnya, penting untuk memastikan bahwa Anda mengajukan pertanyaan yang tepat untuk mendapatkan nilai paling besar (informasi yang mengarah pada perbaikan yang dapat ditindaklanjuti) dari proses tersebut. Pertimbangkan pertanyaan-pertanyaan ini untuk membantu Anda memulai dalam membangun diskusi belajar dari pengalaman Anda:

- Apa insidennya?
- Kapan insiden tersebut pertama kali diidentifikasi?
- Bagaimana insiden tersebut diidentifikasi?
- Sistem apa yang diperingatkan tentang aktivitas tersebut?
- Sistem, layanan, dan data apa yang terlibat?
- Apa yang terjadi secara khusus?
- Apa yang berjalan dengan baik?
- Apa yang tidak berjalan dengan baik?
- Proses atau prosedur mana yang gagal atau gagal diskalakan untuk merespons insiden tersebut?
- Apa yang dapat diperbaiki dalam area-area berikut:
  - Personel
    - Apakah orang-orang yang perlu dihubungi benar-benar sedang lowong dan apakah daftar kontak sudah diperbarui?
    - Apakah orang-orang melewatkan pelatihan atau kemampuan yang diperlukan untuk merespons dan menyelidiki insiden tersebut secara efektif?
    - Apakah sumber daya yang tepat siap dan tersedia?

- Proses
  - Apakah proses dan prosedur diikuti?
  - Apakah proses dan prosedur didokumentasikan dan tersedia untuk (jenis) insiden ini?
  - Apakah proses dan prosedur yang diperlukan tidak tersedia?
  - Apakah perespons dapat memperoleh akses tepat waktu ke informasi yang diperlukan untuk merespons masalah ini?
- Teknologi
  - Apakah sistem peringatan yang ada secara efektif mengidentifikasi dan memperingatkan tentang aktivitas?
  - Bagaimana kita bisa mengurangi waktu deteksi hingga 50%?
  - Apakah peringatan yang ada memerlukan perbaikan atau peringatan baru perlu dibuat untuk (jenis) insiden ini?
  - Apakah alat yang ada memungkinkan penyelidikan (pencarian/analisis) insiden yang efektif?
  - Apa yang dapat dilakukan untuk membantu mengidentifikasi (jenis) insiden ini dengan lebih cepat?
  - Apa yang dapat dilakukan untuk membantu mencegah (jenis) insiden ini terjadi lagi?
  - Siapa yang memiliki rencana perbaikan dan bagaimana Anda akan menguji bahwa rencana tersebut telah diterapkan?
  - Bagaimana garis waktu untuk mengimplementasikan dan menguji pemantauan tambahan atau kontrol dan proses pencegahan?

Daftar ini bukanlah daftar lengkap, melainkan dimaksudkan sebagai titik awal untuk mengidentifikasi kebutuhan organisasi dan bisnis dan bagaimana Anda dapat menganalisisnya agar dapat belajar secara efektif dari insiden dan terus meningkatkan postur keamanan Anda. Yang paling penting adalah memulai dengan menyertakan kerangka kerja belajar dari pengalaman sebagai bagian standar dari proses respons insiden, dokumentasi, dan harapan seluruh pemangku kepentingan.

Sumber daya

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS - Menetapkan kerangka kerja untuk belajar dari insiden
- Panduan NCSC CAF - Belajar dari pengalaman

Keamanan aplikasi

Keamanan aplikasi (AppSec) menjelaskan proses keseluruhan mengenai cara Anda mendesain, membangun, dan menguji karakteristik keamanan dari beban kerja yang Anda kembangkan. Anda harus melatih orang di organisasi Anda dengan baik, memahami karakteristik keamanan dari build Anda dan merilis infrastruktur, serta menggunakan otomatisasi untuk mengidentifkasi masalah keamanan.

Mengadopsi pengujian keamanan aplikasi sebagai bagian dari siklus hidup pengembangan perangkat lunak (SDLC) Anda dan memposting proses rilis membantu memastikan bahwa Anda memiliki mekanisme terstruktur untuk mengidentifikasi, memperbaiki, dan mencegah masalah keamanan aplikasi masuk ke lingkungan produksi Anda.

Metodologi pengembangan aplikasi Anda harus menyertakan kontrol keamanan saat Anda mendesain, membangun, men-deploy, dan mengoperasikan beban kerja Anda. Saat melakukannya, selaraskan proses demi penurunan kecacatan yang terus-menerus dan meminimalkan utang teknis. Misalnya, menggunakan pemodelan ancaman dalam fase desain membantu Anda menemukan kecacatan desain lebih dini, dan kecacatan ini lebih mudah diatasi serta tidak terlalu mahal dibandingkan menunggu dan memperbaikinya nanti.

Biaya dan kompleksitas untuk mengatasi kecacatan biasanya lebih rendah jika Anda masuk ke dalam fase SDLC lebih dini. Cara termudah untuk mengatasi masalah ini adalah mengupayakan agar tidak ada kecacatan dari awal. Oleh karena itu, memulai dengan model ancaman membantu Anda fokus mendapatkan hasil yang tepat dari fase desain. Saat program AppSec Anda berkembang, Anda dapat meningkatkan jumlah pengujian yang dilakukan menggunakan otomatisasi, meningkatkan fidelitas umpan balik ke builder, dan menurunkan waktu yang diperlukan untuk peninjauan keamanan. Semua tindakan ini meningkatkan kualitas perangkat lunak yang Anda bangun, dan meningkatkan kecepatan penyiapan fitur untuk masuk ke produksi.

Panduan implementasi ini fokus pada empat area: organisasi dan budaya, keamanan dari pipeline, keamanan di pipeline, dan manajemen dependensi. Setiap area menyediakan sekumpulan prinsip yang dapat Anda implementasikan, dan menyediakan tampilan menyeluruh mengenai cara Anda mendesain, mengembangkan, membangun, men-deploy, dan mengoperasikan beban kerja.

Di AWS, ada beberapa pendekatan yang dapat Anda gunakan saat menangani program keamanan aplikasi Anda. Beberapa pendekatan ini bergantung pada teknologi, sedangkan yang lain fokus pada orang dan aspek organisasi dari program keamanan aplikasi Anda.

Praktik terbaik

- SEC11-BP01 Pelatihan untuk keamanan aplikasi
- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis
- SEC11-BP03 Lakukan uji penetrasi secara teratur
- SEC11-BP04 Peninjauan kode manual
- SEC11-BP05 Pusatkan layanan untuk paket dan dependensi
- SEC11-BP06 Lakukan deployment perangkat lunak secara terprogram
- SEC11-BP07 Nilai karakteristik keamanan pipeline secara teratur
- SEC11-BP08 Buat program yang menanamkan kepemilikan keamanan dalam tim beban kerja

SEC11-BP01 Pelatihan untuk keamanan aplikasi

Berikan pelatihan kepada builder dalam organisasi Anda mengenai praktik umum untuk pengembangan dan pengoperasian aplikasi yang aman. Adopsi praktik pengembangan yang berfokus pada keamanan akan membantu mengurangi kemungkinan munculnya masalah yang hanya terdeteksi pada tahap peninjauan keamanan.

Hasil yang diinginkan: Perangkat lunak harus didesain dan dikembangkan dengan mempertimbangkan keamanan. Saat builder di sebuah organisasi berlatih praktik pengembangan aman yang dimulai dengan model ancaman, langkah ini meningkatkan keseluruhan kualitas dan keamanan perangkat lunak yang dibuat. Pendekatan ini dapat mempersingkat waktu untuk mengirimkan perangkat lunak atau fitur karena tidak perlu banyak pengerjaan ulang setelah tahap peninjauan keamanan.

Untuk tujuan praktik terbaik ini, pengembangan aman merujuk pada perangkat lunak yang sedang ditulis dan alat atau sistem yang mendukung siklus hidup pengembangan perangkat lunak (SDLC).

Antipola umum:

- Menunggu sampai peninjauan keamanan, lalu mempertimbangkan karakteristik keamanan sistem.
- Menyerahkan semua keputusan keamanan kepada tim keamanan.
- Gagal menyampaikan cara keputusan diambil di SDLC terkait ekspektasi keseluruhan keamanan atau kebijakan organisasi.
- Terlambat melibatkan diri dalam proses peninjauan keamanan.

Manfaat menjalankan praktik terbaik ini:

- Memiliki pengetahuan yang lebih baik seputar persyaratan organisasi untuk keamanan pada fase awal siklus pengembangan.
- Dapat mengidentifikasi dan mengatasi potensi masalah keamanan lebih cepat, sehingga dapat mengirim fitur lebih cepat.
- Peningkatan kualitas perangkat lunak dan sistem.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Sediakan pelatihan kepada builder di organisasi Anda. Memulai dengan kursus pemodelan ancaman adalah dasar yang baik guna membantu pelatihan untuk keamanan. Idealnya, builder harus dapat mengakses secara mandiri informasi yang relevan dengan beban kerja mereka. Akses ini membantu mereka mengambil keputusan yang tepat tentang karakteristik keamanan sistem yang mereka bangun tanpa perlu bertanya kepada tim lain. Proses melibatkan tim keamanan untuk peninjauan harus diatur dengan jelas dan mudah diikuti. Langkah-langkah di proses peninjauan harus disertakan dalam pelatihan keamanan. Jika tersedia templat atau pola implementasi yang diketahui, keduanya harus mudah dicari dan berhubungan dengan persyaratan keamanan keseluruhan. Pertimbangkan untuk menggunakan AWS CloudFormation, AWS Cloud Development Kit (AWS CDK) Constructs, Service Catalog, atau alat pembuat templat lainnya untuk mengurangi kebutuhan akan konfigurasi kustom.

Langkah implementasi

- Mulai latih builder dengan memberikan kursus terkait pemodelan ancaman untuk membangun dasar yang baik, dan bimbing mereka untuk mengetahui cara memikirkan tentang keamanan.
- Berikan akses ke AWS Training dan Sertifikasi, industri, atau pelatihan Partner AWS.
- Berikan pelatihan terkait proses peninjauan keamanan organisasi Anda, yang menguraikan pembagian tanggung jawab antara tim keamanan, tim beban kerja, dan pemegang kepentingan lainnya.
- Publikasikan panduan layanan mandiri terkait cara memenuhi persyaratan keamanan Anda, termasuk templat dan contoh kode, jika tersedia.
- Dapatkan umpan balik secara rutin dari tim builder terkait pengalaman mereka seputar pelatihan dan proses peninjauan keamanan, dan gunakan umpan balik tersebut untuk meningkatkan kualitasnya.

- Gunakan kampanye game day atau bug bash untuk membantu menurunkan jumlah masalah, dan mengasah kemampuan builder Anda.

Sumber daya

Praktik Terbaik Terkait:

- SEC11-BP08 Buat program yang menanamkan kepemilikan keamanan dalam tim beban kerja

Dokumen terkait:

- AWS Training dan Sertifikasi
- Cara berpikir tentang tata kelola keamanan cloud
- Cara memanfaatkan pemodelan ancaman
- Mempercepat pelatihan – AWS Skills Guild

Video terkait:

- Keamanan proaktif: Pertimbangan dan pendekatan

Contoh terkait:

- Lokakarya tentang pemodelan ancaman
- Kesadaran industri untuk developer

Layanan terkait:

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK) (AWS CDK) Constructs
- Service Catalog
- AWS BugBust

SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Otomatiskan pengujian untuk karakteristik keamanan sepanjang siklus hidup pengembangan dan rilis. Otomatisasi mempermudah identifikasi yang konsisten dan berulang atas potensi masalah dalam perangkat lunak sebelum rilis, yang mengurangi risiko masalah keamanan dalam perangkat lunak yang disediakan.

Hasil yang diinginkan: Tujuan pengujian otomatis adalah menyediakan cara terprogram dalam mendeteksi potensi masalah lebih dini dan lebih sering sepanjang siklus hidup pengembangan. Saat Anda mengotomatiskan pengujian regresi, Anda dapat menjalankan kembali pengujian fungsional dan nonfungsional untuk memastikan bahwa perangkat lunak yang diuji sebelumnya masih berfungsi seperti yang diharapkan setelah perubahan. Saat Anda mendefinisikan pengujian unit keamanan untuk memeriksa apakah ada kesalahan konfigurasi umum, seperti autentikasi yang rusak atau hilang, Anda dapat mengidentifikasi dan memperbaiki masalah ini lebih dini dalam proses pengembangan.

Otomatisasi pengujian menggunakan kasus pengujian yang dibuat berdasarkan tujuan untuk validasi aplikasi, berdasarkan persyaratan aplikasi dan fungsionalitas yang diinginkan. Hasil pengujian otomatis berdasarkan perbandingan output pengujian yang dibuat dengan output yang diharapkan, sehingga mempercepat keseluruhan siklus hidup pengujian. Metodologi pengujian seperti pengujian regresi dan rangkaian pengujian unit adalah pilihan yang terbaik untuk otomatisasi. Otomatisasi pengujian karakteristik keamanan memungkinkan builder menerima umpan balik otomatis tanpa harus menunggu peninjauan keamanan. Pengujian otomatis dalam bentuk analisis kode statis atau dinamis dapat meningkatkan kualitas kode dan membantu mendeteksi potensi masalah perangkat lunak lebih dini dalam siklus hidup pengembangan.

Antipola umum:

- Tidak menyampaikan kasus pengujian dan hasil pengujian dari pengujian otomatis.
- Hanya menjalankan pengujian otomatis segera sebelum rilis.
- Mengotomatiskan kasus pengujian dengan berulang kali mengubah persyaratan.
- Gagal memberikan panduan mengenai cara menangani hasil pengujian keamanan.

Manfaat menjalankan praktik terbaik ini:

- Menurunkan dependensi pada orang yang mengevaluasi karakteristik keamanan sistem.

- Memiliki temuan yang konsisten di beberapa aliran kerja meningkatkan konsisten.
- Menurunkan kemungkinan munculnya masalah keamanan dalam produksi perangkat lunak.
- Periode waktu lebih pendek antara deteksi dan penyelesaian karena mengidentifikasi masalah perangkat lunak lebih dini.
- Meningkatkan visibilitas perilaku sistemik atau berulang di beberapa aliran kerja, yang dapat digunakan untuk mendorong peningkatan berskala organisasi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Saat Anda membuat perangkat lunak, adopsi beragam mekanisme untuk pengujian perangkat lunak guna memastikan bahwa Anda menguji aplikasi untuk kedua persyaratan aplikasi, berdasarkan logika bisnis aplikasi, dan persyaratan nonfungsional, yang fokus pada keandalan, performa, dan keamanan aplikasi.

Pengujian keamanan aplikasi statis (SAST) menganalisis kode sumber Anda untuk mendeteksi pola keamanan anomali dan memberikan indikasi untuk kode yang rawan cacat. SAST mengandalkan input statis, seperti dokumentasi (spesifikasi persyaratan, dokumentasi desain, dan spesifikasi desain) dan sumber kode aplikasi untuk menguji beragam masalah keamanan yang diketahui. Penganalisis kode statis dapat membantu mempercepat analisis kode dalam volume besar. NIST Quality Group menyediakan perbandingan Penganalisis Keamanan Kode Sumber, yang menyertakan alat sumber terbuka untuk Pemindai Kode Bita dan Pemindai Kode Biner.

Lengkapi pengujian statis Anda dengan metodologi pengujian keamanan analisis dinamis (DAST), yang menjalankan pengujian terhadap aplikasi yang berjalan untuk mengidentifikasi potensi perilaku yang tidak diharapkan. Pengujian dinamis dapat digunakan untuk mendeteksi potensi masalah yang tidak terdeteksi melalui analisis statis. Pengujian di tahap repositori kode, build, dan pipeline memungkinkan Anda memeriksa berbagai jenis potensi masalah agar tidak masuk ke dalam kode Anda. Amazon CodeWhisperer menyediakan rekomendasi kode, termasuk pemindaian keamanan, di IDE builder. Amazon CodeGuru Reviewer dapat mengidentifikasi masalah penting, masalah keamanan, dan bug yang sulit ditemukan selama pengembangan aplikasi, dan menyediakan rekomendasi untuk meningkatkan kualitas kode.

Lokakarya Keamanan untuk Developer menggunakan alat developer AWS, seperti AWS CodeBuild, AWS CodeCommit, dan AWS CodePipeline, untuk otomatisasi pipeline rilis yang menyertakan metodologi pengujian SAST dan DAST.

Saat Anda menjalani SDLC, buat proses iteratif yang menyertakan peninjauan aplikasi berkala bersama tim keamanan Anda. Umpan balik yang didapatkan dari peninjauan keamanan ini harus diatasi dan divalidasi sebagai bagian dari peninjauan kesiapan rilis Anda. Tinjauan ini membuat postur keamanan aplikasi yang kokoh, dan memberikan umpan balik yang dapat ditindaklanjuti kepada builder untuk menangani potensi masalah.

Langkah implementasi

- Implementasikan IDE yang konsisten, peninjauan kode, dan alat CI/CD yang menyertakan pengujian keamanan.
- Pertimbangkan posisi yang tepat dalam SDLC untuk memblokir pipeline daripada hanya memberi tahu builder bahwa masalah perlu diselesaikan.
- Lokakarya Keamanan untuk Developer memberikan contoh mengintegrasikan pengujian statis dan dinamis ke dalam pipeline rilis.
- Menjalankan pengujian atau analisis kode menggunakan alat otomatis, seperti Amazon CodeWhisperer yang diintegrasikan dengan IDE developer, dan Amazon CodeGuru Reviewer untuk memindai kode dalam penerapan, membantu builder mendapatkan umpan balik pada waktu yang tepat.
- Saat membangun menggunakan AWS Lambda, Anda dapat menggunakan Amazon Inspector untuk memindai kode aplikasi dalam fungsi Anda.
- Lokakarya CI/CD AWS menyediakan titik awal dalam membangun pipeline CI/CD pada AWS.
- Saat pengujian otomatis disertakan dalam pipeline CI/CD, Anda harus menggunakan sistem tiket untuk melacak notifikasi dan penyelesaian masalah perangkat lunak.
- Untuk pengujian keamanan yang mungkin menghasilkan temuan, menautkan ke panduan untuk penyelesaian membantu builder meningkatkan kualitas kode.
- Analisis temuan secara berkala dari alat otomatis untuk memprioritaskan otomatisasi berikutnya, pelatihan builder, atau kampanye kesadaran.

Sumber daya

Dokumen terkait:

- Pengiriman Berkelanjutan dan Deployment Berkelanjutan
- Partner Kompetensi DevOps AWS

- Partner Kompetensi Keamanan AWS untuk Keamanan Aplikasi
- Memilih pendekatan CI/CD Well-Architected
- Memantau peristiwa CodeCommit di Amazon EventBridge dan Amazon CloudWatch Events
- Deteksi rahasia dalam Peninjauan Amazon CodeGuru
- Percepat deployment di AWS dengan tata kelola yang efektif
- Cara AWS memanfaatkan otomatisasi deployment yang aman tanpa campur tangan

Video terkait:

- Tanpa campur tangan: Mengotomatiskan pipeline pengiriman berkelanjutan di Amazon
- Mengotomatiskan pipeline CI/CD lintas akun

Contoh terkait:

- Kesadaran industri untuk developer
- Tata Kelola AWS CodePipeline (GitHub)
- Lokakarya Keamanan untuk Developer
- Lokakarya CI/CD AWS

SEC11-BP03 Lakukan uji penetrasi secara teratur

Lakukan uji penetrasi perangkat lunak secara teratur. Mekanisme ini membantu mengidentifikasi potensi masalah perangkat lunak yang tidak dapat dideteksi oleh pengujian otomatis atau peninjauan kode manual. Mekanisme ini juga membantu Anda memahami efikasi kontrol deteksi Anda. Uji penetrasi harus mencoba untuk menentukan apakah perangkat lunak dapat dibuat untuk berkinerja dengan cara-cara yang tak terduga, seperti mengungkapkan data yang seharusnya dilindungi, atau memberikan izin yang lebih luas daripada yang diharapkan.

Hasil yang diinginkan: Uji penetrasi digunakan untuk mendeteksi, menyelesaikan, dan memvalidasi karakteristik keamanan aplikasi Anda. Uji penetrasi yang teratur dan terjadwal harus dilakukan sebagai bagian dari siklus hidup pengembangan perangkat lunak (SDLC). Temuan dari uji penetrasi harus diatasi sebelum perangkat lunak dirilis. Anda harus menganalisis temuan dari uji penetrasi untuk mengidentifikasi apakah ada masalah yang dapat ditemukan menggunakan otomatisasi.

Memiliki uji penetrasi yang teratur dan dapat diulangi serta menyertakan mekanisme umpan balik yang aktif membantu menginformasikan panduan kepada builder dan meningkatkan kualitas perangkat lunak.

Antipola umum:

- Hanya melakukan uji penetrasi untuk masalah keamanan yang diketahui atau umum.
- Melakukan uji penetrasi aplikasi tanpa pustaka dan alat pihak ketiga yang dependen.
- Hanya melakukan uji penetrasi untuk masalah keamanan paket, dan tidak mengevaluasi logika bisnis yang diimplementasikan.

Manfaat menjalankan praktik terbaik ini:

- Peningkatan kredibilitas karakteristik keamanan dari perangkat lunak sebelum rilis.
- Peluang untuk mengidentifikasi pola aplikasi yang dipilih, yang menghasilkan kualitas perangkat lunak yang lebih baik.
- Loop umpan balik yang mengidentifikasi lebih dini di siklus pengembangan di mana otomatisasi atau pelatihan tambahan dapat meningkatkan karakteristik keamanan perangkat lunak.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Uji penetrasi adalah langkah pengujian keamanan terstruktur untuk menjalankan skenario pelanggaran keamanan terencana guna mendeteksi, menyelesaikan, dan memvalidasi kontrol keamanan. Uji penetrasi dimulai dengan pengintaian, yang mana data dikumpulkan berdasarkan desain aplikasi dan dependensinya saat ini. Daftar kurasi skenario pengujian khusus keamanan dibuat dan dijalankan. Tujuan utama pengujian ini adalah mengungkap masalah keamanan di aplikasi Anda, yang dapat dieksploitasi untuk mendapatkan akses yang tidak direncanakan ke lingkungan Anda, atau akses yang tidak diotorisasi ke data Anda. Anda harus melakukan uji penetrasi saat meluncurkan fitur baru atau setiap kali aplikasi Anda menjalani perubahan besar pada implementasi teknis atau fungsi.

Anda harus mengidentifikasi tahap yang paling sesuai dalam siklus hidup pengembangan untuk melakukan uji penetrasi. Pengujian ini harus dilakukan pada waktu yang hampir mendekati status rilis fungsionalitas sistem yang direncanakan, tetapi ada waktu yang cukup untuk menyelesaikan masalah yang ada.

Langkah implementasi

- Miliki proses terstruktur mengenai cara uji penetrasi dicakup. Merancang proses berdasarkan model ancaman adalah cara yang baik dalam mempertahankan konteks.
- Identifikasi tempat yang sesuai dalam siklus pengembangan untuk melakukan uji penetrasi. Hal ini harus dilakukan saat ada perubahan minim yang diharapkan pada aplikasi, tetapi ada waktu yang cukup untuk melakukan penyelesaian masalah.
- Latih builder Anda untuk mengetahui apa saja yang diharapkan dari temuan uji penetrasi dan cara mendapatkan informasi dalam penyelesaian.
- Gunakan alat untuk mempercepat alat uji penetrasi dengan mengotomatiskan pengujian yang umum atau dapat diulang.
- Analisis temuan uji penetrasi untuk mengidentifikasi masalah keamanan sistemik, dan gunakan data ini untuk menginformasikan pengujian tambahan yang diotomatisasi dan pendidikan builder yang sedang berlangsung.

Sumber daya

Praktik Terbaik Terkait:

- SEC11-BP01 Pelatihan untuk keamanan aplikasi
- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Uji Penetrasi AWS menyediakan panduan mendetail untuk uji penetrasi pada AWS
- Percepat deployment di AWS dengan tata kelola yang efektif
- Partner Kompetensi Keamanan AWS
- Modernisasi arsitektur uji penetrasi Anda di AWS Fargate
- Simulator injeksi Kesalahan AWS

Contoh terkait:

- Otomatiskan pengujian API dengan AWS CodePipeline (GitHub)
- Pembantu keamanan yang diotomatiskan (GitHub)

SEC11-BP04 Peninjauan kode manual

Lakukan peninjauan kode manual atas perangkat lunak yang Anda hasilkan. Proses ini membantu memverifikasi bahwa orang yang menulis kode bukan satu-satunya orang yang memeriksa kualitas kode.

Hasil yang diinginkan: Menyertakan langkah peninjauan kode manual selama pengembangan meningkatkan kualitas perangkat lunak yang ditulis, membantu mengasah kemampuan anggota tim yang kurang berpengalaman, dan memberikan peluang untuk mengidentifikasi titik yang cocok untuk otomatisasi. Peninjauan kode manual dapat didukung oleh pengujian dan alat otomatis.

Antipola umum:

- Tidak melakukan peninjauan kode sebelum deployment.
- Penulis dan peninjau kode adalah orang yang sama.
- Tidak menggunakan otomatisasi untuk membantu atau mengatur peninjauan kode.
- Tidak melatih builder agar memahami keamanan aplikasi sebelum mereka meninjau kode.

Manfaat menjalankan praktik terbaik ini:

- Peningkatan kualitas kode.
- Peningkatan konsistensi pengembangan kode sepanjang penggunaan ulang pendekatan umum.
- Penurunan jumlah masalah yang ditemukan selama uji penetrasi dan tahap-tahap terakhir.
- Peningkatan transfer ilmu di dalam tim.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Langkah peninjauan harus diimplementasikan sebagai bagian dari keseluruhan alur manajemen kode. Spesifikasinya bergantung pada pendekatan yang digunakan untuk pencabangan, permintaan penarikan, dan penggabungan. Anda mungkin menggunakan AWS CodeCommit atau solusi pihak ketiga seperti GitHub, GitLab, atau Bitbucket. Apa pun metode yang Anda gunakan, penting untuk memastikan bahwa proses Anda memerlukan peninjauan kode sebelum di-deploy di lingkungan produksi. Menggunakan alat seperti Amazon CodeGuru Reviewer dapat mempermudah pengaturan proses peninjauan kode.

Langkah implementasi

- Implementasikan langkah peninjauan manual sebagai bagian dari alur manajemen kode Anda dan lakukan peninjauan ini sebelum melanjutkan.
- Pertimbangkan Amazon CodeGuru Reviewer untuk mengelola dan membantu dalam peninjauan kode.
- Implementasikan alur persetujuan yang mengharuskan peninjauan kode selesai sebelum kode dapat lanjut ke tahap berikutnya.
- Pastikan ada proses untuk mengidentifikasi masalah yang ditemukan selama peninjauan kode manual yang dapat dideteksi secara otomatis.
- Integrasikan langkah peninjauan kode manual menggunakan cara yang selaras dengan praktik pengembangan kode Anda.

Sumber daya

Praktik Terbaik Terkait:

- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Mengerjakan permintaan penarikan di repositori AWS CodeCommit
- Mengerjakan templat aturan persetujuan di AWS CodeCommit
- Tentang permintaan penarikan di GitHub
- Otomatiskan peninjauan kode dengan Amazon CodeGuru Reviewer
- Mengotomatisasi deteksi kerentanan keamanan dan bug di pipeline CI/CD menggunakan CLI Amazon CodeGuru Reviewer

Video terkait:

- Peningkatan berkelanjutan kualitas kode dengan Amazon CodeGuru

Contoh terkait:

- Lokakarya Keamanan untuk Developer

SEC11-BP05 Pusatkan layanan untuk paket dan dependensi

Berikan layanan terpusat agar tim builder dapat memperoleh paket perangkat lunak dan dependensi lainnya. Hal ini memungkinkan validasi paket sebelum paket disertakan dalam perangkat lunak yang Anda tulis, dan memberikan sumber data untuk analisis perangkat lunak yang digunakan dalam organisasi Anda.

Hasil yang diinginkan: Perangkat lunak yang terdiri dari sekumpulan paket perangkat lunak lainnya selain kode yang ditulis. Ini mempermudah untuk menggunakan implementasi fungsionalitas yang berulang kali digunakan, seperti pengurai JSON atau pustaka enkripsi. Memusatkan secara logis sumber untuk paket dan dependensi ini memberikan mekanisme bagi tim keamanan untuk memvalidasi karakteristik paket sebelum paket digunakan. Pendekatan ini juga menurunkan risiko masalah tidak terduga yang disebabkan oleh perubahan dalam paket yang ada, atau oleh tim builder, termasuk paket arbitrer langsung dari internet. Gunakan pendekatan ini sehubungan dengan alur pengujian manual dan otomatis untuk meningkatkan kredibilitas kualitas perangkat lunak yang dikembangkan.

Antipola umum:

- Menarik paket dari repositori arbitrer di internet.
- Tidak menguji paket baru sebelum menyediakannya kepada builder.

Manfaat menjalankan praktik terbaik ini:

- Pemahaman lebih baik mengenai paket apa yang digunakan di perangkat lunak yang dibangun.
- Dapat memberi tahu tim beban kerja saat paket perlu diperbarui berdasarkan pemahaman siapa yang menggunakan apa.
- Menurunkan risiko penyertaan paket bermasalah di perangkat lunak Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Sedang

Panduan implementasi

Sediakan layanan terpusat untuk paket dan dependensi dengan cara yang mudah digunakan bagi builder. Layanan terpusat dapat dipusatkan secara logis daripada diimplementasikan sebagai sistem monolitik. Pendekatan ini memungkinkan Anda menyediakan layanan dengan cara yang memenuhi kebutuhan builder Anda. Anda harus mengimplementasikan cara yang efisien untuk menambahkan

paket ke repositori saat pembaruan terjadi atau persyaratan baru muncul. Layanan AWS seperti AWS CodeArtifact atau solusi partner AWS serupa menyediakan cara untuk mengirim kapabilitas ini.

Langkah Implementasi:

- Implementasikan layanan repositori terpusat secara logis yang tersedia di semua lingkungan tempat perangkat lunak dikembangkan.
- Sertakan akses ke repositori sebagai bagian dari proses vending Akun AWS.
- Buat otomatisasi untuk menguji paket sebelum paket dipublikasikan ke repositori.
- Pertahankan metrik paket yang paling sering digunakan, bahasa, dan tim dengan jumlah perubahan tertinggi.
- Sediakan mekanisme otomatis untuk tim builder guna meminta paket baru dan memberikan umpan balik.
- Pindai paket secara rutin di repositori Anda untuk mengidentifikasi potensi dampak masalah yang baru ditemukan.

Sumber daya

Praktik Terbaik Terkait:

- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Percepat deployment di AWS dengan tata kelola yang efektif
- Perketat keamanan paket Anda dengan toolkit CodeArtifact Package Origin Control
- Mendeteksi masalah keamanan dalam pencatatan log dengan Amazon CodeGuru Reviewer
- Level rantai Pasokan untuk Artefak Perangkat Lunak (SLSA)

Video terkait:

- Keamanan proaktif: Pertimbangan dan pendekatan
- Filosofi Keamanan AWS (re:Invent 2017)
- Saat keamanan, keselamatan, dan urgensi menjadi penting: Menangani Log4Shell

Contoh terkait:

- Pipeline Publikasi Paket Beberapa Wilayah (GitHub)
- Memublikasikan Modul Node.js di AWS CodeArtifact menggunakan AWS CodePipeline (GitHub)
- Sampel Pipeline AWS CDK Java CodeArtifact (GitHub)
- Distribusikan paket NuGet .NET pribadi dengan AWS CodeArtifact (GitHub)

SEC11-BP06 Lakukan deployment perangkat lunak secara terprogram

Lakukan deployment perangkat lunak secara terprogram jika memungkinkan. Pendekatan ini mengurangi kemungkinan terjadinya kegagalan deployment atau masalah tak terduga karena kesalahan manusia.

Hasil yang diinginkan: Meminimalkan campur tangan manusia dari data adalah prinsip utama pengembangan yang aman di AWS Cloud. Prinsip ini termasuk cara Anda melakukan deployment pada perangkat lunak.

Dengan tidak bergantung pada orang untuk men-deploy perangkat lunak, Anda akan mendapatkan manfaat peningkatan kredibilitas bahwa apa yang Anda uji adalah apa yang di-deploy, dan deployment dilakukan secara konsisten setiap kali dijalankan. Perangkat lunak tidak perlu diubah agar berfungsi di lingkungan yang berbeda. Menggunakan prinsip pengembangan aplikasi dua belas faktor, terutama eksternalisasi konfigurasi, memungkinkan Anda men-deploy kode yang sama ke beberapa lingkungan tanpa memerlukan perubahan. Menandatangani paket perangkat lunak secara kriptografis adalah cara yang baik untuk memastikan bahwa tidak ada yang berubah di antara lingkungan. Keseluruhan hasil dari pendekatan ini adalah penurunan risiko proses perubahan dan peningkatan konsistensi rilis perangkat lunak.

Antipola umum:

- Men-deploy perangkat lunak secara manual ke tahap produksi.
- Melakukan perubahan secara manual ke perangkat lunak agar dapat menyesuaikan dengan lingkungan yang berbeda.

Manfaat menjalankan praktik terbaik ini:

- Peningkatan kredibilitas dalam proses rilis perangkat lunak.

- Penurunan risiko kegagalan perubahan yang berdampak pada fungsionalitas bisnis.
- Peningkatan jadwal rilis karena risiko terhadap perubahan lebih rendah.
- Kapabilitas pengembalian (rollback) otomatis untuk peristiwa tidak terduga selama deployment.
- Kemampuan untuk membuktikan secara kriptografis bahwa perangkat lunak yang diuji adalah perangkat lunak yang di-deploy.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Buat struktur Akun AWS Anda untuk menghapus akses manusia yang persisten dari lingkungan dan gunakan alat CI/CD untuk melakukan deployment. Rancang aplikasi Anda sehingga data konfigurasi khusus lingkungan diperoleh dari sumber eksternal, seperti AWS Systems Manager Parameter Store. Tanda tangani paket setelah paket diuji, dan validasikan tanda tangan ini selama deployment. Konfigurasikan pipeline CI/CD Anda untuk mendorong kode aplikasi dan menggunakan canary untuk mengonfirmasi deployment yang berhasil. Gunakan alat seperti AWS CloudFormation atau AWS CDK untuk menentukan infrastruktur Anda, lalu gunakan AWS CodeBuild dan AWS CodePipeline untuk melakukan operasi CI/CD.

Langkah implementasi

- Bangun pipeline CI/CD yang ditetapkan dengan baik untuk menyederhanakan proses deployment.
- Menggunakan AWS CodeBuild dan AWS Code Pipeline untuk menyediakan kemampuan CI/CD mempermudah untuk mengintegrasikan pengujian keamanan ke jalur Anda.
- Ikuti panduan pemisahan lingkungan di laporan resmi Mengatur Lingkungan AWS Anda Menggunakan Beberapa Akun.
- Pastikan tidak ada akses manusia yang persisten ke lingkungan tempat beban kerja produksi berjalan.
- Rancang aplikasi Anda untuk mendukung eksternalisasi data konfigurasi.
- Pertimbangkan untuk men-deploy menggunakan model deployment blue/green.
- Implementasikan canary untuk memvalidasi deployment perangkat lunak yang berhasil.
- Gunakan alat kriptografis seperti AWS Signer atau AWS Key Management Service (AWS KMS) untuk menandatangani dan memverifikasi paket perangkat lunak yang Anda deploy.

Sumber daya

Praktik Terbaik Terkait:

- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Lokakarya CI/CD AWS
- Percepat deployment di AWS dengan tata kelola yang efektif
- Mengotomatiskan deployment aman tanpa campur tangan
- Penandatanganan kode menggunakan AWS Certificate Manager Private CA dan kunci asimetris AWS Key Management Service
- Penandatanganan Kode, Kontrol Integritas dan Kepercayaan untuk AWS Lambda

Video terkait:

- Tanpa campur tangan: Mengotomatiskan pipeline pengiriman berkelanjutan di Amazon

Contoh terkait:

- Deployment Blue/Green dengan AWS Fargate

SEC11-BP07 Nilai karakteristik keamanan pipeline secara teratur

Terapkan prinsip Pilar Keamanan Well-Architected pada pipeline Anda, dengan perhatian khusus pada pemisahan izin. Nilai karakteristik keamanan infrastruktur pipeline Anda secara teratur. Mengelola keamanan pipeline secara efektif akan memungkinkan Anda memberikan keamanan perangkat lunak yang lolos melalui pipeline.

Hasil yang diinginkan: Pipeline yang digunakan untuk membangun dan men-deploy perangkat lunak Anda harus mengikuti rekomendasi praktik yang sama seperti beban kerja lainnya di lingkungan Anda. Pengujian yang diimplementasikan di pipeline seharusnya tidak dapat diedit oleh builder yang menggunakannya. Pipeline seharusnya hanya memiliki izin yang diperlukan untuk deployment yang dilakukannya dan harus mengimplementasikan perlindungan untuk menghindari deployment ke

lingkungan yang salah. Pipeline tidak boleh bergantung pada kredensial jangka panjang, dan harus dikonfigurasikan untuk memberikan status sehingga integritas lingkungan build dapat divalidasi.

Antipola umum:

- Pengujian keamanan yang dapat dilewati oleh builder.
- Izin yang terlalu luas untuk pipeline deployment.
- Pipeline tidak dikonfigurasikan untuk memvalidasi input.
- Tidak rutin meninjau izin yang terkait dengan infrastruktur CI/CD Anda.
- Penggunaan kredensial jangka panjang atau yang diberi hardcode.

Manfaat menjalankan praktik terbaik ini:

- Kredibilitas lebih tinggi pada integritas perangkat lunak yang dibangun dan di-deploy melalui pipeline.
- Kemampuan untuk menghentikan deployment saat ada aktivitas yang mencurigakan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Memulai dengan layanan CI/CD terkelola yang mendukung peran IAM menurunkan risiko kebocoran kredensial. Menerapkan prinsip Pilar Keamanan ke infrastruktur pipeline CI/CD Anda dapat membantu Anda menentukan titik mana yang dapat ditingkatkan keamanannya. Mengikuti Arsitektur Rujukan Pipeline Deployment AWS adalah titik awal yang baik untuk membangun lingkungan CI/CD Anda. Meninjau secara rutin implementasi pipeline dan menganalisis log untuk menemukan perilaku tidak terduga dapat membantu Anda memahami pola penggunaan pipeline yang digunakan untuk men-deploy perangkat lunak.

Langkah implementasi

- Mulai dengan Arsitektur Rujukan Pipeline Deployment AWS.
- Pertimbangkan untuk menggunakan AWS IAM Access Analyzer agar dapat secara terprogram membuat kebijakan IAM hak akses paling rendah untuk pipeline.
- Integrasikan pipeline Anda dengan pemantauan dan peringatan sehingga Anda mendapatkan notifikasi aktivitas tidak terduga atau tidak normal, untuk layanan terkelola AWS, Amazon

EventBridge memungkinkan Anda merutekan data ke target seperti AWS Lambda atau Amazon Simple Notification Service (Amazon SNS).

Sumber daya

Dokumen terkait:

- Arsitektur Rujukan Pipeline Deployment AWS
- Pemantauan AWS CodePipeline
- Praktik terbaik keamanan untuk AWS CodePipeline

Contoh terkait:

- Dasbor pemantauan DevOps (GitHub)

SEC11-BP08 Buat program yang menanamkan kepemilikan keamanan dalam tim beban kerja

Buat program atau mekanisme yang memberdayakan tim builder untuk membuat keputusan keamanan tanpa perangkat lunak yang mereka buat. Tim keamanan Anda masih harus memvalidasi keputusan ini selama peninjauan, tetapi menanamkan kepemilikan keamanan dalam tim builder memungkinkan beban kerja dibangun dengan lebih cepat dan lebih aman. Mekanisme ini juga mendukung budaya kepemilikan yang secara positif memengaruhi operasi sistem yang Anda buat.

Hasil yang diinginkan: Untuk menanamkan kepemilikan keamanan dan pengambilan keputusan dalam tim builder, Anda dapat melatih builder mengenai cara berpikir tentang keamanan atau meningkatkan pelatihan mereka bersama orang keamanan yang diikutsertakan atau dikaitkan dengan tim builder. Pendekatan mana pun bisa dilakukan dan memungkinkan tim mengambil keputusan keamanan yang lebih berkualitas pada awal siklus pengembangan. Model kepemilikan ini didasarkan pada pelatihan untuk keamanan aplikasi. Memulai dengan model ancaman untuk beban kerja tertentu akan membantu fokus pada pemikiran desain dalam konteks yang sesuai. Manfaat lain dalam memiliki komunitas builder yang fokus pada keamanan, atau kelompok rekayasawan keamanan yang bekerja sama dengan tim builder, adalah pemahaman yang lebih mendalam mengenai cara perangkat lunak ditulis. Pemahaman ini membantu Anda menentukan area peningkatan selanjutnya dalam kemampuan otomatisasi Anda.

Antipola umum:

- Menyerahkan semua keputusan desain keamanan kepada tim keamanan.
- Tidak menangani persyaratan keamanan cukup dini dalam proses pengembangan.
- Tidak memperoleh umpan balik dari builder dan orang keamanan dalam pengoperasian program.

Manfaat menjalankan praktik terbaik ini:

- Waktu penyelesaian peninjauan keamanan lebih cepat.
- Penurunan masalah keamanan yang hanya terdeteksi pada tahap peninjauan keamanan.
- Peningkatan keseluruhan kualitas perangkat lunak yang ditulis.
- Peluang untuk mengidentifikasi dan memahami masalah sistemik atau area dari peningkatan bernilai tinggi.
- Penurunan jumlah pengerjaan ulang yang diperlukan akibat temuan dari peninjauan keamanan.
- Peningkatan persepsi fungsi keamanan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Rendah

Panduan implementasi

Mulai dengan panduan di SEC11-BP01 Pelatihan untuk keamanan aplikasi. Lalu, identifikasi model operasional untuk program yang menurut Anda paling sesuai dengan organisasi Anda. Dua pola utamanya adalah melatih builder atau menyertakan orang keamanan ke dalam tim builder. Setelah Anda memutuskan pendekatan awal, Anda harus melakukan uji coba dengan satu grup atau grup kecil tim beban kerja untuk membuktikan model mana yang sesuai dengan organisasi Anda. Dukungan kepemimpinan dari bagian builder dan keamanan organisasi membantu pengiriman dan kesuksesan program. Saat Anda membangun program, penting untuk memilih metrik yang dapat digunakan untuk menunjukkan nilai program. Belajar dari cara AWS menangani masalah ini adalah pengalaman pembelajaran yang baik. Praktik terbaik ini sangat berfokus pada budaya dan perubahan organisasi. Alat yang Anda gunakan harus mendukung kolaborasi antara komunitas keamanan dan builder.

Langkah implementasi

- Mulai dengan melatih builder Anda untuk memahami keamanan aplikasi.
- Buat komunitas dan program orientasi untuk mengedukasi builder.

- Pilih nama untuk program. Pelindung, Jawara, atau Pendukung adalah nama yang sering digunakan.
- Identifikasi model yang akan digunakan: latih builder, sertakan rekayasawan keamanan, atau miliki peran keamanan afinitas.
- Identifikasi sponsor proyek dari grup keamanan, builder, dan grup lain yang berpotensi.
- Lacak metrik untuk jumlah orang yang terlibat dalam program, waktu yang dihabiskan untuk peninjauan, dan umpan balik dari orang keamanan dan builder. Gunakan metrik ini untuk membuat peningkatan.

Sumber daya

Praktik Terbaik Terkait:

- SEC11-BP01 Pelatihan untuk keamanan aplikasi
- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Cara memanfaatkan pemodelan ancaman
- Cara berpikir tentang tata kelola keamanan cloud

Video terkait:

- Keamanan proaktif: Pertimbangan dan pendekatan

Kesimpulan

Keamanan merupakan upaya yang berkelanjutan. Ketika insiden terjadi, insiden harus diperlakukan sebagai peluang untuk meningkatkan keamanan arsitektur. Memiliki kontrol identitas yang kuat, mengotomatiskan respons terhadap peristiwa keamanan, melindungi infrastruktur di beberapa tingkat, dan mengelola data yang terklasifikasi baik dengan enkripsi akan memberikan pertahanan mendalam yang harus diimplementasikan setiap organisasi. Upaya ini lebih mudah berkat fungsi pemrograman dan layanan serta fitur AWS yang dibahas dalam laporan ini.

AWS berusaha membantu Anda membangun dan mengoperasikan arsitektur yang melindungi informasi, sistem, dan aset sekaligus memberikan nilai bisnis.

Kontributor

Individu dan organisasi berikut ini memiliki kontribusi dalam dokumen ini:

- Sarita Dharankar, Security Pillar Lead, Well-Architected, Amazon Web Services
- Adam Cerini, Senior Solution Architect, Amazon Web Services
- Bill Shinn, Senior Principal, Kantor CISO, Amazon Web Services
- Brigid Johnson, Senior Software Development Manager, AWS Identity, Amazon Web Services
- Byron Pogson, Senior Solution Architect, Amazon Web Services
- Charlie Hammell, Principal Enterprise Architect, Amazon Web Services
- Darran Boyd, Principal Security Solutions Architect, Layanan Keuangan, Amazon Web Services
- Dave Walker, Principal Specialist Solutions Architect, Keamanan dan Kepatuhan, Amazon Web Services
- John Formento, Senior Solution Architect, Amazon Web Services
- Paul Hawkins, Principal, Kantor CISO, Amazon Web Services
- Sam Elmalak, Senior Technology Leader, Amazon Web Services
- Pat Gaw, Principal Security Consultant, Amazon Web Services
- Daniel Begimher, Senior Consultant, Keamanan, Amazon Web Services
- Danny Cortegaca, Senior Security Solutions Architect, Amazon Web Services
- Ana Malhotra, Security Solutions Architect, Amazon Web Services
- Debashis Das, Principal, Kantor CISO, Amazon Web Services
- Reef Dsouza, Principal Solutions Architect, Amazon Web Services
- Brad Burnett, Security Solutions Architect, Identitas, Amazon Web Services
- Anna McAbee, Senior Security Solutions Architect, Deteksi Ancaman dan Respons Insiden, Amazon Web Services
- Jason Garman, Principal Security Solutinos Architect, Amazon Web Services

Bacaan lebih lanjut

Untuk bantuan tambahan, pelajari sumber berikut:

- Laporan resmi Kerangka Kerja AWS Well-Architected
- Pusat Arsitektur AWS

Revisi Dokumen

Berlangganan umpan RSS untuk memperoleh pemberitahuan tentang pembaruan laporan resmi ini.

Perubahan	Deskripsi	Tanggal
Panduan praktik terbaik yang diperbarui	Praktik-praktik terbaik telah diperbarui dengan panduan baru di area-area berikut: Mengoperasikan beban kerja Anda dengan aman dan Melindungi data bergerak.	December 6, 2023
Panduan praktik terbaik yang diperbarui	Pembaruan besar untuk panduan dan praktik terbaik dalam Respons insiden. Beberapa praktik terbaik diperbarui dalam Persiapan. Dua area baru ditambahkan ke Respons insiden: Operasi dan Aktivitas pascainsiden. Praktik terbaik baru SEC10-BP08 Menetapkan kerangka kerja untuk belajar dari insiden ditambahkan.	October 3, 2023
Panduan praktik terbaik yang diperbarui	Praktik terbaik diperbarui dengan panduan baru di area-area berikut: Siapkan dan Simulasikan.	July 13, 2023
Pembaruan untuk Kerangka Kerja baru.	Praktik terbaik diperbarui dengan panduan preskript if dan praktik terbaik baru ditambahkan. Area praktik terbaik baru Keamanan	April 10, 2023

	Aplikasi (AppSec) ditambahkan.	
Laporan resmi diperbarui	Praktik terbaik diperbarui dengan panduan implementasi baru.	December 15, 2022
Laporan resmi diperbarui	Praktik terbaik diperluas dan rencana pengembangan ditambahkan.	October 20, 2022
Pembaruan kecil	Informasi IAM diperbarui untuk menggambarkan praktik terbaik saat ini.	June 28, 2022
Pembaruan kecil	Informasi AWS PrivateLink tambahan ditambahkan dan tautan yang bermasalah dikoreksi.	May 19, 2022
Pembaruan kecil	AWS PrivateLink ditambahkan.	May 6, 2022
Pembaruan kecil	Bahasa non-inklusif dihilangkan.	April 22, 2022
Pembaruan kecil	Informasi tentang VPC Network Access Analyzer ditambahkan.	February 2, 2022
Pembaruan kecil	Penambahan Pilar Pelestarian Lingkungan ke pengantar.	December 2, 2021
Pembaruan kecil	Tautan yang bermasalah diperbaiki.	May 27, 2021
Pembaruan kecil	Perubahan editorial di seluruh dokumen.	May 17, 2021

Pembaruan besar	Bagian tentang tata kelola ditambahkan, detail untuk berbagai bagian ditambahkan, fitur dan layanan baru ditambahkan di seluruh dokumen.	May 7, 2021
Pembaruan kecil	Tautan diperbarui.	March 10, 2021
Pembaruan kecil	Tautan yang bermasalah diperbaiki.	July 15, 2020
Pembaruan untuk Kerangka Kerja baru	Pembaruan panduan untuk akun, identitas, dan manajemen izin.	July 8, 2020
Pembaruan untuk Kerangka Kerja baru	Pembaruan untuk perluasan perangkat di setiap area, praktik terbaik baru, layanan dan fitur.	April 30, 2020
Laporan resmi diperbarui	Pembaruan untuk menggambarkan fitur dan layanan AWS baru, dan pembaruan referensi.	July 1, 2018
Laporan resmi diperbarui	Pembaruan bagian Pemeliharaan dan Konfigurasi Keamanan Sistem untuk menggambarkan fitur dan layanan AWS baru.	May 1, 2017
Publikasi awal	Pilar Keamanan - Kerangka Kerja AWS Well-Architected diterbitkan.	November 1, 2016

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya disediakan sebagai informasi, (b) berisi praktik dan penawaran produk AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tindak menjadi komitmen atau jaminan apa pun dari AWS dan afiliasi, pemasok, atau pemberi lisensinya. Layanan atau produk AWS diberikan “apa adanya” tanpa jaminan, pernyataan, atau syarat apa pun, baik secara tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.