Kerangka Kerja AWS Well-Architected

Pilar Keamanan

Pilar Keamanan: Kerangka Kerja AWS Well-Architected

Table of Contents

Pilar Keamanan - Kerangka Kerja AWS Well-Architected

Tanggal publikasi: 6 November 2024 (Revisi dokumen)

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

Pengantar

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

Enam pilar landasan kerangka kerja:

- Keunggulan Operasional
- Keamanan
- Keandalan
- Efisiensi Kinerja
- Pengoptimalan Biaya
- Keberlanjutan

Laporan ini berfokus pada pilar keamanan. Laporan ini akan membantu Anda memenuhi persyaratan bisnis dan peraturan dengan mengikuti saran-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, developer, dan anggota tim operasi.

Setelah membaca laporan ini, Anda akan memahami saran dan strategi AWS terkini yang bisa digunakan ketika merancang arsitektur cloud dengan mempertimbangkan keamanan. Laporan ini tidak menyediakan detail implementasi atau pola-pola yang berkaitan dengan arsitektur, tetapi

menyertakan referensi ke sumber daya yang relevan untuk informasi ini. Dengan mengadopsi praktik-praktik yang diuraikan dalam laporan ini, Anda dapat membangun arsitektur yang dapat melindungi data dan sistem Anda, mengontrol akses, dan merespons peristiwa-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:

- Implementasikan 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.
- Menjaga 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.
- Terapkan 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).
- Lakukan otomatisasi 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.
- Lindungi data bergerak dan data diam: Klasifikasikan data sesuai tingkatan sensitivitasnya dan mekanisme pengunaannya, seperti enkripsi, tokenisasi dan kontrol akses jika sesuai.
- Jauhkan keterlibatan manusia dalam penanganan data: Gunakan mekanisme dan alat untuk mengurangi atau menghilangkan 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

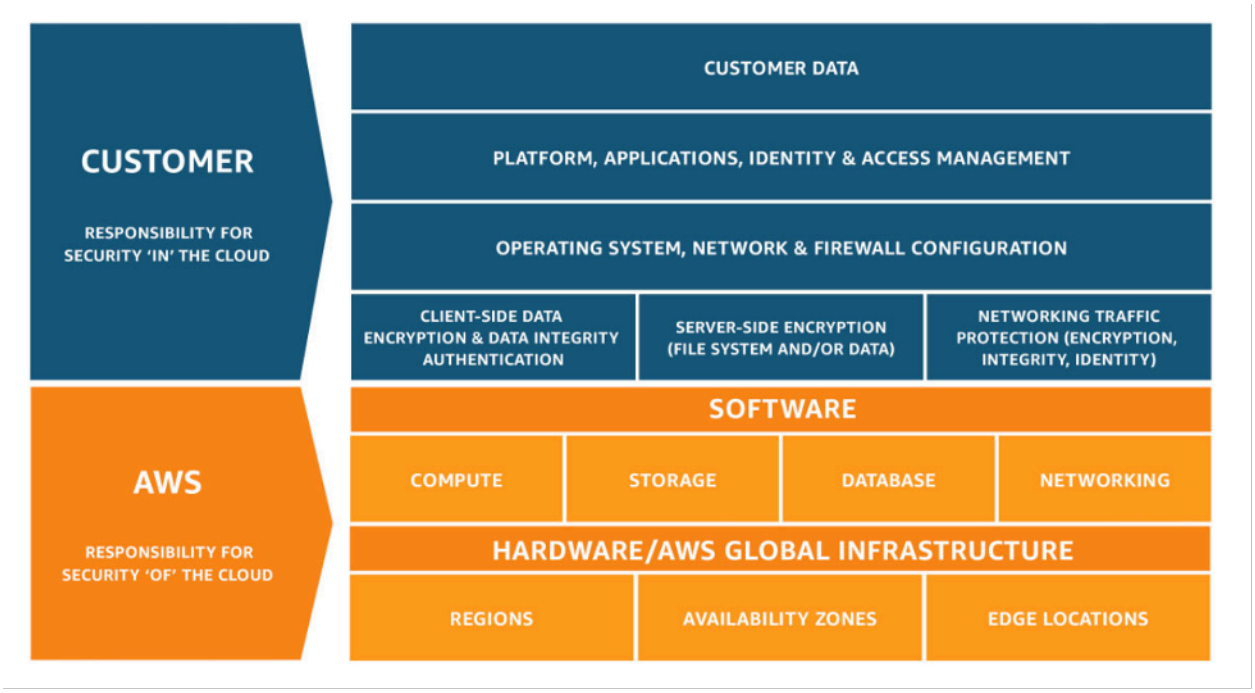
Tanggung jawab bersama

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

Tanggung jawab “Keamanan dalam Cloud” oleh Pelanggan – Tanggung jawab pelanggan akan ditentukan oleh layanan AWS Cloud yang dipilih oleh seorang 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 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 seorang 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 termasuk:

- 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 Pelanggan – Kontrol yang sepenuhnya merupakan tanggung jawab pelanggan berdasarkan aplikasi yang mereka deploy dalam layanan AWS. Contohnya termasuk:

- 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. Sumber daya yang bermanfaat 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 sebuah penyedia identitas seperti AWS IAM Identity Center, dan kontrol-kontrol detektif 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-kontrol detektif. 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 yang melekat. 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.

Likelihood	Risk Level				
Very Likely	Low	Medium	High	Critical	Critical
Likely	Low	Medium	Medium	High	Critical
Possible	Low	Low	Medium	Medium	High
Unlikely	Low	Low	Medium	Medium	High
Very unlikely	Low	Low		Medium	High
Consequence	Minimal	Low	Medium	High	Severe

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.

Kelola akun secara terpusat: AWS Organizations mengotomatiskan pembuatan dan pengelolaan AWS akun dan kontrol 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. Organisasi akan memungkinkan Anda untuk mengelompokkan akun ke dalam unit organisasi (OU), yang dapat mewakili lingkungan yang berbeda berdasarkan persyaratan dan tujuan beban kerja.

Atur 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 akan 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. Ini akan mengotomatiskan pengaturan akun di Organisasi AWS Anda, mengotomatiskan penyediaan, menerapkan pagar pembatas (yang mencakup pencegahan dan deteksi), dan memberi Anda dasbor untuk visibilitas.

Konfigurasikan layanan dan sumber daya secara terpusat: AWS Organizations akan membantu Anda mengonfigurasi layanan-layanan AWS yang berlaku untuk semua akun Anda. Misalnya, Anda dapat mengonfigurasi pencatatan log terpusat untuk semua tindakan yang dilakukan di organisasi Anda menggunakan AWS CloudTrail, dan mencegah akun anggota dari menonaktifkan pencatatan log. Anda juga dapat mengumpulkan data secara terpusat untuk aturan-aturan yang telah Anda tetapkan dengan menggunakan AWS Config, sehingga Anda dapat mengaudit kepatuhan beban kerja Anda dan bereaksi cepat terhadap perubahan. AWS CloudFormation StackSets dapat memudahkan Anda dalam mengelola tumpukan AWS CloudFormation secara terpusat di berbagai akun dan OU yang ada 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 di antara lingkungan (seperti produksi, pengembangan, dan pengujian) dan beban kerja melalui strategi multi-akun. 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 terkait, dan lingkungan ke dalam akun terpisah, meningkatkan keamanan di seluruh infrastruktur cloud.

Anti-pola 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 tersentralisasi 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 diterapkan: Tinggi

Panduan implementasi

Akun AWS memberikan suatu batasan isolasi keamanan di antara beban kerja atau sumber daya yang beroperasi pada tingkat sensitivitas yang berbeda. AWS menyediakan alat-alat untuk mengelola beban kerja cloud Anda dalam skala yang sesuai melalui strategi multi-akun untuk memanfaatkan batasan isolasi ini. Untuk panduan tentang konsep, pola, dan implementasi strategi multi-akun pada AWS, lihat Mengatur Lingkungan AWS Anda dengan 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, sehingga 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 merupakan dua layanan yang dapat Anda gunakan untuk menerapkan dan mengelola struktur multi-akun ini di lingkungan AWS Anda. AWS Organizations memungkinkan Anda untuk mengatur akun ke dalam hierarki yang ditentukan oleh satu atau beberapa lapisan OU, dengan setiap OU yang berisi beberapa akun anggota. Kebijakan kontrol layanan (SCP) memungkinkan administrator organisasi untuk membuat kontrol

pencegahan terperinci pada akun anggota, dan AWS Config dapat digunakan untuk membuat kontrol-kontrol proaktif dan detektif pada akun anggota. Banyak layanan AWS terintegrasi dengan AWS Organizations untuk menyediakan kontrol administratif yang didelegasikan dan melakukan tugas khusus layanan di semua akun anggota dalam organisasi.

Berlapis di atas AWS Organizations, AWS Control Tower menyediakan pengaturan praktik terbaik satu kali klik untuk lingkungan AWS banyak akun dengan zona landasan. Zona landasan adalah titik masuk ke lingkungan multi-akun yang dibuat oleh Control Tower. Control Tower memberikan beberapa manfaat dibanding AWS Organizations. Tiga manfaat yang memberikan tata kelola akun yang lebih baik adalah:

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

Langkah-langkah implementasi

1. Merancang struktur unit organisasi: Struktur unit organisasi yang dirancang dengan baik mengurangi beban manajemen yang diperlukan untuk membuat dan memelihara kebijakan kontrol layanan dan kontrol keamanan lainnya. Struktur unit organisasi Anda harus selaras dengan kebutuhan bisnis Anda, sensitivitas data, dan struktur beban kerja.
2. Buat zona landasan untuk lingkungan multi-akun Anda: zona landasan menyediakan fondasi keamanan dan infrastruktur yang konsisten dari mana organisasi Anda dapat dengan cepat mengembangkan, meluncurkan, dan menerapkan beban kerja. Anda dapat menggunakan zona landasan atau AWS Control Tower yang dibuat khusus untuk mengatur lingkungan Anda.
3. Tetapkan pagar pembatas: Terapkan pagar pembatas keamanan yang konsisten untuk lingkungan Anda melalui zona landasan Anda. AWS Control Tower menyediakan daftar kontrol wajib dan opsional yang dapat digunakan. 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 akan disebarkan ke Wilayah yang Anda aktifkan. 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 melakukan deploy sumber daya termasuk kebijakan, peran dand grup IAM ke dalam Akun AWS and Wilayah yang berbeda dari templat yang disetujui.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC02-BP04 Mengandalkan penyedia identitas tersentralisasi

Dokumen terkait:

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

Video terkait:

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

SEC01-BP02 Mengamankan properti dan pengguna root akun

Pengguna root adalah pengguna yang memiliki hak istimewa paling banyak dalam sebuah Akun AWS, dengan akses administratif penuh ke semua sumber daya di dalam akun, dan dalam beberapa

kasus tidak dapat dibatasi oleh kebijakan keamanan. Melakukan deaktivasi 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 diinginkan: Mengamankan pengguna root membantu mengurangi kemungkinan kerusakan yang tidak disengaja atau disengaja dapat terjadi melalui penyalahgunaan kredensial pengguna root. Menerapkan kontrol-kontrol detektif juga dapat memberikan peringatan kepada personel yang tepat saat ada tindakan dilakukan menggunakan pengguna root.

Anti-pola umum:

- Menggunakan pengguna root untuk tugas selain yang memerlukan kredensial pengguna root.
- Tidak melakukan pengujian terhadap rencana-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: Mengamankan akses ke pengguna root akan membangun keyakinan bahwa tindakan-tindakan yang dilakukan di akun Anda sudah dikontrol dan diaudit.

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

Panduan implementasi

AWS menawarkan banyak alat untuk membantu mengamankan akun Anda. Namun, karena beberapa tindakan ini tidak dinyalakan 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 yang dilakukan untuk menilai dan memantau 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 Anda gunakan untuk membuat akun. Karena peningkatan akses yang diberikan kepada pengguna root AWS, Anda harus membatasi penggunaan pengguna root AWS untuk melakukan tugas-tugas yang secara khusus memerlukannya. Kredensial masuk pengguna root harus diamankan secara ketat, dan selalu gunakan autentikasi multi-faktor (MFA) untuk pengguna root Akun AWS.

Selain alur autentikasi normal untuk masuk log in ke pengguna root Anda yang menggunakan nama pengguna, kata sandi, perangkat autentikasi multi-faktor (MFA), ada alur pemulihan akun untuk masuk ke pengguna root Akun AWS Anda yang mendapatkan akses ke alamat email dan nomor telepon yang dikaitkan 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, Anda juga perlu mempertimbangkan 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 satu 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-langkah implementasi

Langkah-langkah implementasi berikut direkomendasikan untuk membuat kontrol bagi pengguna root tersebut. Jika berlaku, rekomendasi direferensikan silang ke benchmark CIS AWS Foundations versi 1.4.0. Selain langkah-langkah ini, konsultasikan pedoman praktik terbaik AWS untuk mengamankan Akun AWS dan sumber daya Anda.

Kontrol pencegahan

1. Siapkan informasi kontak yang akurat untuk akun tersebut.
    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 melakukan transisi proses menggunakan kunci tersebut untuk menggunakan kunci akses sementara dari peran (IAM) AWS Identity and Access Management, 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 tersebut akan ditetapkan ke nilai acak yang tidak akan ditampilkan kepada Anda. Pertimbangkan untuk menggunakan alur pengaturan ulang kata sandi dari akun manajemen Organisasi AWS Anda untuk mendapatkan akses ke akun anggota jika diperlukan.
   b. Untuk Akun AWS atau akun AWS Organization manajemen terpisah, Anda disarankan untuk membuat dan menyimpan kredensial dengan aman untuk pengguna root. Gunakan MFA untuk pengguna root.
4. Aktifkan kontrol pencegahan untuk pengguna root akun anggota di lingkungan multi-akun AWS.
   a. Pertimbangkan untuk menggunakan pagar penjaga pencegahan Jangan Izinkan Pembuatan Kunci Akses Root untuk Pengguna Root untuk akun anggota.
   b. Pertimbangkan untuk menggunakan pagar penjaga pencegahan Jangan Izinkan Tindakan sebagai Pengguna Root untuk akun anggota.
5. Jika Anda memerlukan kredensial untuk pengguna root:
   a. Gunakan kata sandi yang kompleks.
   b. Nyalakan 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 mengkonfigurasi MFA untuk pengguna root, ikuti instruksi untuk membuat MFA virtual atau perangkat perangkat keras MFA.

d. Pertimbangkan untuk mendaftarkan beberapa perangkat MFA untuk cadangan. Hingga 8 perangkat MFA diperbolehkan per akun.
   - Perhatikan bahwa mendaftarkan lebih dari satu perangkat MFA untuk pengguna root secara otomatis mematikan 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 bahwa kata sandi yang diketahui orang lain secara tidak sengaja dapat disalahgunakan.

Kontrol detektif

- Buat alarm untuk mendeteksi penggunaan kredensial root (CIS 1.7). Amazon GuardDuty dapat memantau dan memperingatkan penggunaan kredensial API pengguna root melalui temuan RootCredentialUsage.
- Mengevaluasi dan menerapkan kontrol-kontrol detektif yang termasuk dalam paket kesesuaian Pilar Keamanan yang Dirancang dengan Baik AWS untuk AWS Config, atau jika menggunakan AWS Control Tower, kontrol yang sangat disarankan tersedia di 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 tersebut tidak boleh digunakan untuk tugas sehari-hari, bahkan tugas administratif. Hanya masuk sebagai pengguna root untuk melakukan tugas-tugas AWS yang

membutuhkan 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 di bawah Kontak Alternatif berfungsi. Pantau kotak masuk email untuk melihat apakah ada notifikasi keamanan yang Anda terima dari <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-praktik terbaik di bagian Respons Insiden di laporan resmi Pilar Keamanan untuk mendapatkan informasi lebih lanjut tentang cara membangun strategi respons insiden untuk Akun AWS Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP01 Memisahkan beban kerja menggunakan akun
- SEC02-BP01 Gunakan 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
- Pedoman 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 break glass pada AWS

- 10 item keamanan teratas untuk ditingkatkan dalam Akun AWS Anda
- Apa yang harus saya lakukan jika saya melihat aktivitas tanpa izin dalam Akun AWS saya?

Video terkait:

- Aktifkan adopsi AWS dalam skala besar dengan menggunakan 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 IAM AWS

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 playbook yang jelas, tetap ada risiko ketidakkonsistenan ketika orang melakukan tugas berulang. Ini dapat terjadi terutama ketika orang-orang memiliki tanggung jawab yang beragam dan mereka harus memberikan respons terhadap peringatan yang belum dikenal. 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 tersentralisasi 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 Terus ikuti info terbaru tentang ancaman dan rekomendasi keamanan
- SEC01-BP05 Kurangi cakupan manajemen keamanan
- SEC01-BP06 Otomatiskan deployment kontrol keamanan standar
- SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi dengan menggunakan sebuah model ancaman
- SEC01-BP08 Evaluasi dan implementasikan 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 kontrol dan tujuan kontrol yang perlu Anda terapkan pada beban kerja Anda. Validasi berkelanjutan terhadap kontrol dan tujuan kontrol dapat membantu Anda mengukur efektivitas mitigasi risiko.

Hasil yang diinginkan: Tujuan dari kontrol keamanan terhadap bisnis Anda didefinisikan dengan baik dan selaras dengan persyaratan kepatuhan Anda. Kontrol diimplementasikan dan diberlakukan melalui otomatisasi dan kebijakan serta terus dievaluasi untuk mengetahui efektivitasnya dalam mencapai tujuan Anda. Bukti efektivitas pada suatu titik waktu dan selama periode waktu tertentu dapat mudah dilaporkan kepada auditor.

Anti-pola umum:

- Persyaratan peraturan, ekspektasi pasar, dan standar industri untuk keamanan yang dapat dijamin belum dipahami dengan baik untuk bisnis Anda
- Kerangka kerja keamanan siber dan tujuan kontrol Anda tidak selaras dengan persyaratan bisnis Anda
- Implementasi kontrol tidak sepenuhnya selaras dengan tujuan kontrol Anda secara terukur
- Anda tidak menggunakan otomatisasi untuk melaporkan efektivitas kontrol Anda

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

Panduan implementasi

Ada banyak kerangka kerja keamanan siber umum yang bisa menjadi dasar untuk tujuan kontrol keamanan Anda. Pertimbangkan persyaratan peraturan, ekspektasi pasar, dan standar industri untuk bisnis Anda guna menentukan kerangka kerja mana yang paling memenuhi kebutuhan Anda. Contohnya termasuk AICPA SOC 2, HITRUST, PCI-DSS, ISO 27001, dan NIST SP 800-53.

Untuk tujuan kontrol yang Anda identifikasi, pahami cara layanan AWS yang Anda gunakan membantu Anda mencapai tujuan tersebut. Gunakan AWS Artifactuntuk menemukan dokumentasi dan laporan yang selaras dengan kerangka kerja target Anda yang menjelaskan ruang lingkup tanggung jawab yang dicakup oleh AWS dan panduan untuk ruang lingkup yang tersisa yang menjadi tanggung jawab Anda. Untuk panduan khusus layanan lebih lanjut saat selaras dengan berbagai pernyataan kontrol kerangka kerja, lihat Panduan Kepatuhan Pelanggan AWS.

Saat Anda menentukan kontrol yang memenuhi tujuan Anda, lakukan kodifikasi pemberlakuan menggunakan kontrol preventif, dan lakukan otomatisasi mitigasi dengan menggunakan kontrol-kontrol detektif. Bantu mencegah konfigurasi dan tindakan sumber daya yang tidak sesuai di seluruh kebijakan kontrol layanan (SCP) yang Anda AWS Organizations gunakan. Terapkan aturan AWS Configuntuk memantau dan melaporkan sumber daya yang tidak sesuai, kemudian beralih aturan ke model penegakan setelah yakin dengan perilakunya. Untuk menerapkan set aturan yang telah ditentukan dan dikelola yang selaras dengan kerangka kerja keamanan siber Anda, evaluasi penggunaan standar AWS Security Hub sebagai opsi pertama Anda. Standar AWS Foundational Service Best Practices (FSBP) dan CIS AWS Foundations Benchmark adalah titik awal yang baik dengan kontrol-kontrol yang selaras dengan banyak tujuan yang tercakup dalam berbagai kerangka kerja standar. Jika Security Hub tidak secara intrinsik memiliki deteksi kontrol yang diinginkan, maka dapat dilengkapi dengan menggunakan paket kesesuaian.AWS Config

Gunakan Paket Mitra APN yang direkomendasikan oleh tim AWS Global Security and Compliance Acceleration (GSCA) untuk mendapatkan bantuan dari penasihat keamanan, lembaga konsultasi, sistem pengumpulan dan pelaporan bukti, auditor, dan layanan pelengkap lainnya bila diperlukan.

Langkah-langkah implementasi

1. Evaluasi kerangka kerja keamanan siber umum, dan selaraskan tujuan kontrol Anda dengan kerangka kerja yang dipilih.
2. Dapatkan dokumentasi yang relevan tentang panduan dan tanggung jawab untuk kerangka kerja Anda menggunakan AWS Artifact. Pahami bagian kepatuhan mana yang termasuk dalam porsi AWS pada model tanggung jawab bersama dan bagian mana yang merupakan tanggung jawab Anda.
3. Gunakan SCP, kebijakan sumber daya, kebijakan kepercayaan peran, dan pagar pembatas lainnya untuk mencegah konfigurasi dan tindakan sumber daya yang tidak mematuhi persyaratan.
4. Lakukan evaluasi terhadap penerapan standar Security Hub dan paket kesesuaian AWS Config yang selaras dengan tujuan kontrol Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC03-BP01 Menetapkan persyaratan akses
- SEC04-BP01 Mengonfigurasi pencatatan log layanan dan aplikasi
- SEC07-BP01 Pahami skema klasifikasi data Anda

- OPS01-BP03 Evaluasi persyaratan tata kelola
- OPS01-BP04 Evaluasi persyaratan kepatuhan
- PERF01-BP05 Menggunakan kebijakan dan arsitektur referensi
- COST02-BP01 Mengembangkan kebijakan berdasarkan keperluan organisasi Anda

Dokumen terkait:

- Panduan Kepatuhan Pelanggan AWS

Alat terkait:

- AWS Artifact

SEC01-BP04 Terus ikuti info terbaru tentang ancaman dan rekomendasi keamanan

Terus ikuti perkembangan ancaman dan mitigasi terbaru dengan memantau publikasi intelijen dan umpan data ancaman industri untuk mendapatkan pemberitahuan. Evaluasi penawaran layanan terkelola yang diperbarui secara otomatis berdasarkan data ancaman terbaru.

Hasil yang diinginkan: Anda tetap mendapat informasi karena publikasi industri diperbarui dengan ancaman dan rekomendasi terbaru. Anda menggunakan otomatisasi untuk mendeteksi potensi kerentanan dan paparan saat Anda mengidentifikasi ancaman baru. Anda mengambil tindakan mitigasi terhadap ancaman tersebut. Anda mengadopsi layanan AWS yang diperbarui secara otomatis dengan intelijen ancaman terbaru.

Anti-pola umum:

- Tidak memiliki mekanisme yang andal dan dapat diulangi untuk terus mendapatkan informasi tentang intelijen ancaman terbaru.
- Memelihara inventaris manual berisi portofolio teknologi, beban kerja, dan dependensi Anda yang memerlukan peninjauan oleh manusia untuk menemukan potensi kerentanan dan paparan.
- Tidak memiliki mekanisme untuk memperbarui beban kerja dan dependensi Anda ke versi terkini yang tersedia yang memberikan mitigasi ancaman yang diketahui.

Manfaat menjalankan praktik terbaik ini: Menggunakan sumber intelijen ancaman untuk tetap up to date akan mengurangi risiko kehilangan perubahan penting pada lanskap ancaman yang dapat memengaruhi bisnis Anda. Penerapan otomatisasi untuk melakukan pemindaian, deteksi, dan remediasi jika ada potensi kerentanan atau paparan dalam beban kerja Anda serta dependensinya dapat membantu Anda memitigasi risiko secara cepat dan terprediksi, dibandingkan dengan alternatif manual. Hal ini membantu mengontrol waktu dan biaya yang terkait dengan mitigasi kerentanan.

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

Panduan implementasi

Tinjau publikasi intelijen ancaman tepercaya untuk terus mengikuti perkembangan lanskap ancaman. Konsultasikan dengan basis pengetahuan MITRE ATT&CK untuk dokumentasi tentang taktik, teknik, dan prosedur (TTP) adversarial yang diketahui. Tinjau daftar Kerentanan dan Paparan Umum (CVE) MITRE untuk tetap mendapat informasi tentang kerentanan yang diketahui dalam produk yang Anda andalkan. Pahami risiko kritis terhadap aplikasi web dengan 10 Proyek OWASP paling populer Open Worldwide Application Security Project (OWASP).

Selalu dapatkan kabar terbaru tentang peristiwa keamanan AWS dan langkah-langkah perbaikan yang disarankan dengan Buletin Keamanan AWS untuk CVE.

Untuk mengurangi tenaga dan biaya secara keseluruhan dalam mengikuti perkembangan terbaru, pertimbangkan menggunakan layanan AWS yang secara otomatis menambahkan intelijen ancaman baru dari waktu ke waktu. Misalnya, Amazon GuardDuty selalu dapatkan kabar terbaru dengan kecerdasan menghadapi ancaman industri untuk mendeteksi perilaku anomali dan tanda ancaman dalam akun Anda. Amazon Inspector secara otomatis menyimpan basis data CVE yang digunakannya untuk fitur pemindaian berkelanjutan yang diperbarui. Baik AWS WAF maupun AWS Shield Advanced menyediakan grup aturan terkelola yang diperbarui secara otomatis saat ancaman baru muncul.

Kaji pilar keunggulan operasional Well-Architected untuk manajemen dan penambalan armada otomatis.

Langkah-langkah implementasi

- Berlangganlah ke pemberitahuan untuk publikasi intelijen ancaman yang relevan dengan bisnis dan industri Anda. Berlangganlah ke Buletin Keamanan AWS.
- Pertimbangkan untuk mengadopsi layanan yang menambahkan kecerdasan menghadapi ancaman baru secara otomatis, seperti Amazon GuardDuty dan Amazon Inspector.

- Lakukan deployment strategi manajemen dan patching armada yang selaras dengan praktik terbaik Pilar Keunggulan Operasional Well-Architected.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi dengan menggunakan sebuah model ancaman
- OPS01-BP05 Mengevaluasi lanskap ancaman
- OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan

SEC01-BP05 Kurangi cakupan manajemen keamanan

Tentukan apakah Anda dapat mengurangi cakupan keamanan dengan menggunakan layanan AWS yang mengalihkan manajemen kontrol tertentu ke AWS (layanan terkelola). Layanan ini dapat membantu mengurangi tugas pemeliharaan keamanan Anda, seperti penyediaan infrastruktur, penyiapan perangkat lunak, patching, atau pencadangan.

Hasil yang diinginkan: Anda mempertimbangkan ruang lingkup manajemen keamanan Anda saat memilih layanan AWS untuk beban kerja Anda. Biaya overhead manajemen dan tugas pemeliharaan (total biaya kepemilikan, atau TCO) ditimbang terhadap biaya layanan yang Anda pilih, selain pertimbangan Well-Architected lainnya. Anda memasukkan dokumentasi kontrol dan kepatuhan AWS dalam prosedur evaluasi dan verifikasi kontrol Anda.

Anti-pola umum:

- Melakukan deployment beban kerja tanpa sepenuhnya memahami model tanggung jawab bersama untuk layanan yang Anda pilih.
- Meng-host basis data dan teknologi lainnya pada mesin virtual tanpa mengevaluasi sarana yang setara dengan layanan terkelola.
- Tidak menyertakan tugas manajemen keamanan ke dalam total biaya kepemilikan untuk meng-host teknologi pada mesin virtual jika dibandingkan dengan opsi layanan terkelola.

Manfaat menjalankan praktik terbaik ini: Menggunakan layanan terkelola akan dapat mengurangi beban keseluruhan Anda dalam mengelola kontrol keamanan operasional, hal ini dapat mengurangi

risiko keamanan dan total biaya kepemilikan Anda. Waktu yang seharusnya dihabiskan untuk tugas-tugas keamanan tertentu dapat diinvestasikan kembali ke tugas-tugas yang memberikan nilai lebih bagi bisnis Anda. Layanan terkelola juga dapat mengurangi cakupan persyaratan kepatuhan Anda dengan mengalihkan sebagian persyaratan kontrol ke AWS.

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

Panduan implementasi

Ada beberapa cara untuk mengintegrasikan komponen beban kerja Anda di AWS. Untuk menginstal dan menjalankan teknologi pada instans Amazon EC2, Anda sering kali harus memikul porsi tanggung jawab keamanan terbesar secara keseluruhan. Untuk membantu mengurangi beban pengoperasian kontrol tertentu, identifikasikan layanan terkelola AWS yang mengurangi cakupan model tanggung jawab bersama Anda dan pahami cara Anda dapat menggunakannya dalam arsitektur yang ada. Contohnya termasuk menggunakan Amazon Relational Database Service (Amazon RDS) untuk menggunakan basis data, Amazon Elastic Kubernetes Service (Amazon EKS) atau Amazon Elastic Container Service (Amazon ECS) untuk mengatur kontainer, atau menggunakan opsi nirserver. Saat membuat aplikasi baru, pikirkan layanan mana yang dapat membantu mengurangi waktu dan biaya dalam mengimplementasikan dan mengelola kontrol keamanan.

Persyaratan kepatuhan juga dapat menjadi faktor ketika memilih layanan. Layanan terkelola dapat mengalihkan sebagian persyaratan yang perlu dipatuhi ke AWS. Diskusikan dengan tim kepatuhan Anda apakah mereka nyaman dengan pengauditan aspek layanan yang Anda operasikan dan kelola serta mau menerima pernyataan kontrol dalam laporan audit AWS yang relevan. Anda dapat memberikan artefak audit yang ditemukan pada AWS Artifact ke auditor atau regulator Anda sebagai bukti kontrol keamanan AWS. Anda juga dapat menggunakan panduan tanggung jawab yang disediakan oleh beberapa artefak audit AWS untuk merancang arsitektur Anda, bersama dengan Panduan Kepatuhan Pelanggan AWS. Panduan ini membantu menentukan kontrol keamanan tambahan yang harus Anda terapkan untuk mendukung kasus penggunaan spesifik di sistem Anda.

Saat menggunakan layanan terkelola, pahami proses dalam memperbarui sumber dayanya ke versi yang lebih baru (misalnya, memperbarui versi basis data yang dikelola oleh Amazon RDS, atau runtime bahasa pemrograman untuk suatu fungsi AWS Lambda). Meskipun layanan terkelola dapat melakukan operasi ini untuk Anda, Anda tetap bertanggung jawab untuk mengonfigurasi waktu pembaruan dan memahami dampaknya pada operasi Anda. Alat seperti AWS Health dapat membantu Anda melacak dan mengelola pembaruan ini di seluruh lingkungan Anda.

Langkah-langkah implementasi

1. Evaluasi komponen beban kerja Anda yang dapat diganti dengan layanan terkelola.
    a. Jika Anda memigrasikan beban kerja ke AWS, pertimbangkan melakukan pengurangan manajemen (waktu dan biaya) dan pengurangan risiko ketika Anda menilai apakah Anda harus meng-host ulang, memfaktor ulang, memplatform ulang, membuat ulang, atau mengganti beban kerja Anda. Terkadang, investasi tambahan pada awal migrasi dapat memiliki penghematan yang signifikan dalam jangka panjang.
2. Pertimbangkan untuk mengimplementasikan layanan terkelola, seperti Amazon RDS, bukan menginstal dan mengelola deployment teknologi Anda sendiri.
3. Gunakan panduan tanggung jawab di AWS Artifact untuk membantu menentukan kontrol keamanan yang harus Anda terapkan untuk beban kerja Anda.
4. Pelihara inventaris terkait sumber daya yang digunakan, serta terus ikuti layanan dan pendekatan terkini untuk mengidentifikasi peluang baru dalam mengurangi cakupan.

Sumber daya

Praktik-praktik terbaik terkait:

- PERF02-BP01 Memilih opsi komputasi terbaik untuk beban kerja Anda
- PERF03-BP01 Menggunakan penyimpanan data yang dibuat khusus yang paling mendukung persyaratan akses data dan penyimpanan data Anda
- SUS05-BP03 Menggunakan layanan terkelola

Dokumen terkait:

- Peristiwa siklus hidup yang direncanakan untuk AWS Health

Alat terkait:

- AWS Health
- AWS Artifact
- Panduan Kepatuhan Pelanggan AWS

Video terkait:

- Bagaimana cara bermigrasi ke instans Amazon RDS atau Aurora MySQL DB menggunakan DMS AWS?
- AWS re:Invent 2023 - Mengelola peristiwa siklus hidup sumber daya sesuai skala dengan AWS Health

SEC01-BP06 Otomatiskan deployment kontrol keamanan standar

Terapkan praktik-praktik DevOps modern saat Anda mengembangkan dan melakukan deployment kontrol keamanan standar di seluruh lingkungan AWS Anda. Tentukan kontrol dan konfigurasi keamanan standar dengan menggunakan templat Infrastruktur sebagai Kode (IaC), catat perubahan dalam sistem kontrol versi, uji perubahan sebagai bagian dari pipeline CI/CD, dan lakukan otomatisasi terhadap deployment perubahan ke lingkungan AWS Anda.

Hasil yang diinginkan: Template IaC merekam kontrol keamanan standar dan memasukkannya ke sistem kontrol versi. Pipeline CI/CD diterapkan untuk mendeteksi perubahan serta melakukan otomatisasi terhadap pengujian dan deployment lingkungan AWS Anda. Pagar pembatas diterapkan untuk mendeteksi dan memperingatkan kesalahan konfigurasi dalam templat sebelum melanjutkan ke deployment. Beban kerja di-deploy ke lingkungan yang menerapkan kontrol standar. Tim memiliki akses untuk melakukan deployment konfigurasi layanan yang disetujui melalui mekanisme mandiri. Strategi pencadangan dan pemulihan yang aman diterapkan untuk konfigurasi kontrol, skrip, dan data terkait.

Anti-pola umum:

- Membuat perubahan pada kontrol keamanan standar Anda secara manual, melalui konsol web atau antarmuka baris perintah.
- Mengandalkan tim beban kerja individual untuk secara manual mengimplementasikan kontrol yang ditentukan oleh tim pusat.
- Mengandalkan tim keamanan pusat untuk melakukan deployment kontrol tingkat beban kerja atas permintaan tim beban kerja.
- Mengizinkan individu atau tim yang sama untuk mengembangkan, menguji, dan melakukan deployment skrip otomatisasi kontrol keamanan tanpa pemisahan tugas atau pemeriksaan dan keseimbangan yang tepat.

Manfaat menjalankan praktik terbaik ini: Menggunakan templat untuk menentukan kontrol keamanan standar Anda akan memungkinkan Anda untuk melacak dan membandingkan perubahan dari waktu

ke waktu dengan menggunakan sistem kontrol versi. Penggunaan otomatisasi untuk menguji dan melakukan deployment perubahan akan menghasilkan standardisasi dan kemampuan prediksi, sehingga meningkatkan peluang deployment yang berhasil dan mengurangi tugas manual yang berulang. Penyediaan mekanisme mandiri bagi tim beban kerja untuk melakukan deployment layanan dan konfigurasi yang disetujui akan mengurangi risiko kesalahan konfigurasi dan kesalahan penggunaan. Hal ini juga membantu mereka memasukkan kontrol lebih awal dalam proses pengembangan.

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

Panduan implementasi

Saat mengikuti praktik yang dijelaskan dalam SEC01-BP01 Memisahkan beban kerja menggunakan akun, Anda akan mendapatkan beberapa Akun AWS untuk lingkungan berbeda yang Anda kelola gunakan AWS Organizations. Meskipun masing-masing lingkungan dan beban kerja ini mungkin memerlukan kontrol keamanan yang berbeda, Anda dapat menstandardisasi beberapa kontrol keamanan di seluruh organisasi Anda. Contohnya termasuk mengintegrasikan penyedia identitas tersentralisasi, menentukan jaringan dan firewall, serta mengonfigurasi lokasi standar untuk menyimpan dan menganalisis log. Dengan cara yang sama Anda dapat menggunakan infrastruktur sebagai kode (IAc) untuk menerapkan ketelitian pengembangan kode aplikasi yang sama untuk penyediaan infrastruktur, Anda dapat menggunakan IAc untuk menentukan dan menerapkan kontrol keamanan standar Anda juga.

Jika memungkinkan, tentukan kontrol keamanan Anda dengan cara deklaratif, seperti di AWS CloudFormation, dan simpan dalam sistem kontrol sumber. Gunakan praktik DevOps untuk mengotomatiskan penerapan kontrol Anda untuk rilis yang lebih dapat diprediksi, pengujian otomatis menggunakan alat seperti AWS CloudFormation Guard, dan mendeteksi penyimpangan antara kontrol yang Anda gunakan dan konfigurasi yang Anda inginkan. Anda dapat menggunakan layanan seperti AWS CodePipeline, AWS CodeBuild, dan AWS CodeDeploy untuk membangun konsep pipeline CI/CD. Pertimbangkan panduan dalam Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun untuk mengonfigurasi layanan ini di akun mereka sendiri yang terpisah dari pipeline deployment lainnya.

Anda juga dapat menentukan templat untuk menstandardisasi penentuan dan deployment Akun AWS, layanan, dan konfigurasi. Teknik ini memungkinkan tim keamanan pusat mengelola penentuan ini dan menyediakannya kepada tim beban kerja melalui pendekatan mandiri. Salah satu cara untuk mencapai hal ini adalah dengan menggunakan Service Catalog, di mana Anda dapat memublikasikan templat sebagai produk yang dapat digabungkan oleh tim beban kerja ke dalam

deployment pipeline mereka sendiri. Jika Anda menggunakan AWS Control Tower, beberapa templat dan kontrol tersedia sebagai titik awal. Control Tower juga menyediakan kemampuan Account Factory, sehingga memungkinkan tim beban kerja membuat Akun AWS yang baru menggunakan standar yang Anda tentukan. Kemampuan ini membantu meniadakan dependensi pada tim pusat untuk menyetujui dan membuat akun baru ketika akun tersebut diperlukan oleh tim beban kerja Anda. Anda mungkin memerlukan akun ini untuk mengisolasi komponen beban kerja yang berbeda berdasarkan berbagai alasan, seperti fungsi yang diberikan, sensitivitas data yang diproses, atau perilakunya.

Langkah-langkah implementasi

1. Tentukan cara Anda akan menyimpan dan memelihara templat Anda dalam sebuah sistem kontrol versi.
2. Buat pipeline CI/CD untuk menguji dan menerapkan templat Anda. Tentukan pengujian untuk memeriksa adanya kesalahan konfigurasi dan apakah templat tersebut mematuhi standar perusahaan Anda, atau tidak.
3. Buat katalog templat standar bagi tim beban kerja untuk melakukan deployment Akun AWS dan layanan sesuai dengan kebutuhan Anda.
4. Implementasikan strategi pencadangan dan pemulihan yang aman untuk konfigurasi kontrol, skrip, dan data terkait Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP01 Menggunakan kontrol versi
- OPS05-BP04 Menggunakan sistem manajemen build dan deployment
- REL08-BP05 Melakukan deployment perubahan dengan otomatisasi
- SUS06-BP01 Mengadopsi metode yang dapat menghadirkan peningkatan keberlanjutan dengan cepat

Dokumen terkait:

- Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun

Contoh terkait:

- Mengotomatiskan pembuatan akun, dan penyediaan sumber daya menggunakan Service Catalog, AWS Organizations, dan AWS Lambda
- Memperkuat pipeline DevOps dan melindungi data dengan AWS Secrets Manager, AWS KMS, dan AWS Certificate Manager

Alat terkait:

- AWS CloudFormation Guard
- Akselerator Zona Landasan di AWS

SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi dengan menggunakan sebuah 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. Anda harus memeriksa kembali dan mempertahankan hal ini dengan mempertimbangkan beban kerja Anda, serta lanskap keamanan yang terus berubah.

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

Panduan implementasi

Apa itu pemodelan ancaman?

“Pemodelan ancaman bekerja untuk mengidentifikasi, berkomunikasi, dan memahami ancaman dan mitigasi dalam konteks melindungi sesuatu yang bernilai.” Pemodelan Ancaman Aplikasi Open Web Application Security Project (OWASP)

Mengapa Anda harus menjadi model ancaman?

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 temuan tentang ancaman dan memitigasinya. 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, yang bertujuan untuk menemukan temuan tentang masalah dan mengatasi masalah tersebut dalam proses desain lebih awal, saat biaya dan upaya mitigasi relatif rendah dibandingkan setelahnya dalam siklus hidup. Pendekatan ini sejalan 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 berbasis ancaman.

Kapan pemodelan ancaman harus 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. Tinjau kembali untuk mempertahankan model ancaman Anda dari waktu ke waktu, termasuk saat terjadi perubahan besar, perubahan dalam lanskap ancaman, atau saat Anda mengadopsi fitur atau layanan baru.

Langkah-langkah implementasi

Bagaimana kita bisa melakukan 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 mengajukan pertanyaan terbuka untuk memberikan struktur pada latihan pemodelan ancaman Anda:

1. Apa yang sedang kita 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 cara paling populer untuk menjawab pertanyaan ini, karena membantu Anda memvisualisasikan apa yang Anda bangun, misalnya, menggunakan diagram alir 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 membuat aplikasi web, sepertinya tidak layak untuk membuang waktu melakukan pemodelan ancaman untuk urutan boot tepercaya pada sistem operasi untuk klien browser karena desain Anda tidak akan dapat memengaruhi hal ini.

2. Apa yang bisa salah?

   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. Membuat daftar ini membutuhkan curah pendapat dan kolaborasi antara semua individu dalam tim Anda dan persona relevan yang terlibat dalam latihan pemodelan ancaman. Anda dapat membantu curah pendapat Anda dengan menggunakan model untuk mengidentifikasi ancaman, seperti STRIDE, yang menyarankan berbagai kategori untuk dievaluasi: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, dan Elevation of privilege. Selain itu, Anda mungkin ingin membantu curah pendapat dengan meninjau daftar dan penelitian yang ada untuk mendapatkan inspirasi, termasuk Top 10 OWASP, Katalog Ancaman HiTrust, dan katalog ancaman organisasi Anda sendiri.

3. Apa yang akan kita lakukan tentang hal itu?

   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 merupakan 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 tanggung jawab bersama yang merupakan bagian dari AWS, Anda akan menemukan bahwa layanan AWS berada dalam lingkup banyak program kepatuhan. Program-program tersebut akan membantu Anda memahami penerapan kontrol yang andal yang diterapkan di AWS untuk menjaga keamanan dan kepatuhan cloud. Laporan audit dari program ini dapat diunduh oleh pelanggan AWS dari 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 bagian keamanan khusus yang memberikan panduan tentang kontrol keamanan untuk 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 bisa berupa hal-hal seperti validasi input, penanganan sesi, dan dan penanganan batas. Fokus pada kode kustom karena sebagian besar kerentanan seringnya terjadi di area ini.

4. Apakah kita melakukan pekerjaan 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. Untuk membahas lebih dalam dan langsung, disarankan agar Anda dan tim Anda menyelesaikan Pemodelan ancaman dengan cara yang tepat untuk kursus pelatihan pembangun atau lokakarya. Selain itu, jika Anda mencari panduan tentang cara mengintegrasikan pemodelan ancaman ke dalam siklus pengembangan aplikasi organisasi Anda, lihat kiriman tentang Cara melakukan pendekatan terhadap pemodelan ancaman di Blog Keamanan AWS.

Komposer Ancaman

Untuk membantu dan memandu Anda dalam melakukan pemodelan ancaman, pertimbangkan menggunakan alat Komposer Ancaman, yang bertujuan untuk mengurangi waktu-ke-nilai saat pemodelan ancaman. Alat ini membantu Anda melakukan hal berikut:

- Tulis pernyataan ancaman yang berguna yang selaras dengan tata bahasa ancaman yang bekerja dalam suatu alur kerja non-linier 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, silakan kunjungi Komposer Ancaman dan beralih ke Contoh Ruang Kerja yang ditentukan sistem.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP03 Identifikasikan dan validasikan tujuan kontrol
- SEC01-BP04 Terus ikuti info terbaru tentang ancaman dan rekomendasi keamanan
- SEC01-BP05 Kurangi cakupan manajemen keamanan
- SEC01-BP08 Evaluasi dan implementasikan fitur serta layanan keamanan baru secara rutin

Dokumen terkait:

- Cara melakukan pendekatan terhadap pemodelan ancaman (Blog Keamanan AWS)
- NIST: Panduan untuk Pemodelan Ancaman Sistem yang Terpusat pada Data

Video terkait:

- AWS Summit ANZ 2021 - Cara melakukan pendekatan terhadap pemodelan ancaman
- AWSSummit ANZ 2022 - Menskalakan keamanan — Optimalkan untuk pengiriman yang cepat dan aman

Pelatihan terkait:

- Pemodelan ancaman dengan cara yang tepat untuk pembangun – Pelatihan mandiri virtual AWS Skill Builder
- Pemodelan ancaman dengan cara yang tepat untuk pembangun – Workshop AWS

Alat terkait:

- Komposer Ancaman

SEC01-BP08 Evaluasi dan implementasikan fitur serta layanan keamanan baru secara rutin

Evaluasi dan implementasikan fitur serta layanan keamanan dari AWS dan Partner AWS yang membantu Anda mengembangkan postur keamanan beban kerja Anda.

Hasil yang diinginkan Anda memiliki praktik standar yang memberi tahu Anda tentang fitur dan layanan baru yang dirilis oleh Mitra AWS dan AWS. Anda mengevaluasi pengaruh kemampuan baru ini terhadap desain kontrol saat ini dan yang baru untuk lingkungan dan beban kerja Anda.

Anti-pola umum:

- Anda tidak berlangganan blog AWS dan umpan RSS untuk mempelajari fitur dan layanan baru yang relevan dengan cepat
- Anda mengandalkan berita dan pemberitahuan tentang layanan dan fitur keamanan dari sumber tangan kedua
- Anda tidak menganjurkan pengguna AWS di organisasi Anda untuk terus mengikuti informasi tentang pemberitahuan keamanan terbaru

Manfaat menjalankan praktik terbaik ini: Ketika Anda selalu menggunakan layanan dan fitur keamanan baru, Anda dapat membuat keputusan-keputusan yang tepat berdasarkan informasi tentang penerapan kontrol di lingkungan cloud dan beban kerja Anda. Sumber-sumber ini akan membantu meningkatkan kesadaran akan lanskap keamanan yang terus berubah dan cara layanan AWS dapat digunakan untuk melindungi terhadap ancaman yang baru dan berkembang.

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

Panduan implementasi

AWS menginformasikan pelanggan tentang layanan dan fitur keamanan baru melalui beberapa saluran:

- Yang Terbaru dari AWS
- Blog Berita AWS
- Blog Keamanan AWS
- Buletin Keamanan AWS
- Ikhtisar dokumentasi AWS

Anda dapat berlangganan topik Pembaruan Fitur Harian AWS menggunakan Amazon Simple Notification Service (Amazon SNS) untuk ringkasan pembaruan harian yang komprehensif. Beberapa layanan keamanan, seperti Amazon GuardDuty dan AWS Security Hub, menyediakan topik SNS mereka sendiri untuk tetap mendapat informasi tentang standar baru, temuan, dan pembaruan lainnya untuk layanan tertentu tersebut.

Layanan dan fitur baru juga diumumkan dan dijelaskan secara rinci selama konferensi, acara, dan webinar yang diselenggarakan di seluruh dunia setiap tahunnya. Catatan khusus adalah konferensi keamanan re:Inforce AWS tahunan dan konferensi re:Invent AWS yang lebih umum. Saluran berita AWS yang disebutkan sebelumnya membagikan pengumuman konferensi ini tentang keamanan dan layanan lainnya, dan Anda dapat melihat sesi diskusi edukasional secara mendalam secara online di Saluran acara AWS di YouTube.

Anda juga dapat bertanya kepada tim Akun AWS Anda tentang pembaruan dan rekomendasi layanan keamanan terbaru. Anda dapat menghubungi tim Anda melalui formulir Dukunga Penjualan jika Anda tidak memiliki informasi kontak langsung mereka. Demikian pula, jika Anda berlangganan Dukungan Perusahaan AWS, Anda akan menerima pembaruan mingguan dari Manajer Akun Teknis (TAM) dan dapat menjadwalkan pertemuan tinjauan rutin dengan mereka.

Langkah-langkah implementasi

1. Berlangganlah berbagai blog dan buletin menggunakan pembaca RSS favorit Anda atau berlangganlah topik SNS Daily Features Updates.
2. Evaluasi acara AWS mana yang perlu dihadiri untuk mempelajari secara langsung tentang fitur dan layanan baru.
3. Jadwalkan rapat dengan tim Akun AWS Anda untuk mengajukan pertanyaan apa pun tentang pembaruan layanan dan fitur keamanan.
4. Pertimbangkan untuk berlangganan Dukungan Perusahaan agar dapat melakukan konsultasi rutin dengan Manajer Akun Teknis (TAM).

Sumber daya

Praktik-praktik terbaik terkait:

- PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia
- COST01-BP07 Mengikuti perkembangan rilisan layanan baru

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

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

- Identitas manusia: Identitas manusia yang memerlukan akses ke lingkungan dan aplikasi AWS Anda dapat dikategorikan ke dalam tiga kelompok: tenaga kerja, pihak ketiga, dan pengguna.

  Kelompok tenaga kerja mencakup administrator, developer, dan operator yang merupakan anggota organisasi Anda. Mereka membutuhkan akses untuk mengelola, membangun, dan mengoperasikan sumber daya AWS Anda.

  Pihak ketiga adalah kolaborator eksternal, seperti kontraktor, vendor, atau partner. Mereka berinteraksi dengan sumber daya AWS Anda sebagai bagian dari kerja sama mereka dengan organisasi Anda.

  Pengguna adalah konsumen aplikasi Anda. Mereka mengakses sumber daya AWS Anda melalui browser web, aplikasi klien, aplikasi seluler, 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 juga mencakup mesin yang berjalan di dalam lingkungan AWS Anda, seperti instans Amazon EC2 atau fungsi AWS Lambda. Anda juga dapat mengelola identitas mesin untuk pihak eksternal, atau mesin di luar AWS, yang memerlukan akses ke lingkungan AWS Anda.

Praktik terbaik

- SEC02-BP01 Gunakan mekanisme masuk yang kuat
- SEC02-BP02 Menggunakan kredensial sementara
- SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman
- SEC02-BP04 Mengandalkan penyedia identitas tersentralisasi
- SEC02-BP05 Melakukan audit dan rotasi kredensial secara berkala
- SEC02-BP06 Manfaatkan grup dan atribut pengguna

SEC02-BP01 Gunakan 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 terungkap 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 terjadinya akses yang tidak diinginkan ke kredensial yang ada di AWS dengan menggunakan mekanisme masuk yang kuat untuk pengguna AWS Identity and Access Management(IAM), pengguna rootAkun AWS, AWS IAM Identity Center, dan penyedia identitas pihak ketiga. Tujuan-tujuan ini dapat dicapai dengan MFA, yang menerapkan kebijakan kata sandi yang kuat, dan mendeteksi perilaku adanya pengguna yang masuk secara tidak wajar.

Anti-pola umum:

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

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

Panduan implementasi

Ada banyak cara yang bisa digunakan identitas manusia untuk masuk ke AWS. Praktik terbaik AWS adalah mengandalkan penyedia identitas tersentralisasi dengan menggunakan federasi (federasi SAML 2.0 langsung antara AWS IAM dan IdP tersentralisasi atau menggunakan Pusat Identitas AWS

IAM) saat melakukan autentikasi ke AWS. Dalam kasus tersebut, buat proses masuk yang aman dengan penyedia identitas Anda atau Microsoft Active Directory.

Saat pertama kali membuka sebuah Akun AWS, Anda memulainya dengan pengguna root Akun AWS. Anda harus hanya menggunakan pengguna root akun untuk mengatur akses bagi para pengguna Anda (dan untuk tugas-tugas yang memerlukan pengguna root). Anda harus mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root akun segera setelah membuka Akun AWS Anda dan mengamankan pengguna root menggunakan panduan praktik terbaik AWS.

Pusat Identitas AWS IAM dirancang untuk pengguna tenaga kerja, dan Anda dapat membuat dan mengelola identitas pengguna dalam layanan dan mengamankan proses masuk dengan MFA. AWS Cognito, di sisi lain, dirancang untuk manajemen identitas pelanggan dan akses (CIAM), yang menyediakan kumpulan pengguna dan penyedia identitas untuk identitas pengguna eksternal dalam aplikasi Anda.

Jika Anda membuat pengguna di Pusat Identitas AWS IAM, amankan proses masuk di layanan tersebut dan aktifkan MFA. Untuk identitas pengguna eksternal, Anda dapat menggunakan kumpulan pengguna Amazon Cognito dan mengamankan proses masuk yang ada di layanan tersebut atau melalui salah satu penyedia identitas yang didukung dalam kumpulan pengguna Amazon Cognito.

Selain itu, untuk pengguna di Pusat Identitas AWS IAM, Anda dapat menggunakan Akses Terverifikasi AWS untuk menyediakan lapisan keamanan tambahan dengan memverifikasi postur identitas dan perangkat pengguna sebelum mereka diberi akses ke sumber daya AWS.

Jika Anda menggunakan pengguna AWS Identity and Access Management (IAM), maka Anda seharusnya mengamankan proses masuk dengan menggunakan IAM.

Anda dapat menggunakan Pusat Identitas AWS IAM dan federasi IAM langsung secara bersamaan untuk mengelola akses ke AWS. Anda dapat menggunakan federasi IAM untuk mengelola akses ke AWS Management Console dan layanan serta Pusat Identitas IAM untuk mengelola akses ke aplikasi bisnis seperti QuickSight atau Amazon Q Business.

Apa pun metode masuknya, Anda harus menerapkan kebijakan masuk yang kuat.

Langkah-langkah implementasi

Berikut ini adalah beberapa rekomendasi mekanisme masuk yang kuat secara umum. Pengaturan aktual yang Anda konfigurasikan harus ditetapkan oleh kebijakan perusahaan Anda atau menggunakan sebuah standar seperti NIST 800-63.

- Wajibkan penggunaan MFA. Praktik terbaik IAM adalah mewajibkan penggunaan MFA untuk identitas dan beban kerja manusia. Pengaktifan MFA akan memberikan lapisan keamanan tambahan yang mengharuskan pengguna memberikan kredensial masuk dan kata sandi sekali pakai (OTP) atau string yang dibuat dan diverifikasi secara kriptografis dari perangkat keras.
- Berlakukan panjang minimum kata sandi, yang merupakan faktor utama dari kekuatan kata sandi.
- Berlakukan kompleksitas kata sandi agar kata sandi tidak mudah ditebak.
- Izinkan pengguna mengubah kata sandi mereka sendiri.
- Buatlah identitas individu, bukan kredensial bersama. Dengan membuat identitas individu, Anda dapat memberikan kredensial keamanan yang unik kepada masing-masing pengguna. Pengguna individu juga menyediakan kemampuan untuk mengaudit aktivitas setiap pengguna.

Rekomendasi Pusat Identitas IAM:

- Pusat Identitas IAM menyediakan kebijakan kata sandi yang telah ditentukan sebelumnya saat menggunakan direktori default yang menetapkan panjang kata sandi, kompleksitas, dan persyaratan penggunaan kembali.
- Nyalakan MFA dan konfigurasikan pengaturan sadar konteks atau selalu aktif untuk MFA saat sumber identitasnya adalah direktori default, AWS Managed Microsoft AD, atau AD Connector.
- Izinkan para pengguna untuk mendaftarkan perangkat MFA mereka sendiri.

Rekomendasi direktori kumpulan pengguna Amazon Cognito:

- Konfigurasikan pengaturan Kekuatan kata sandi.
- Wajibkan MFA untuk pengguna.
- Gunakan pengaturan keamanan lanjutan kumpulan pengguna Amazon Cognito untuk fitur seperti autentikasi adaptif yang dapat memblokir setiap upaya masuk yang mencurigakan.

Rekomendasi pengguna IAM:

- Idealnya, Anda menggunakan Pusat Identitas IAM atau federasi langsung. Namun demikian, Anda mungkin membutuhkan pengguna IAM. Dalam hal ini, tetapkan sebuah kebijakan kata sandi untuk pengguna IAM. Anda dapat menggunakan kebijakan kata sandi tersebut untuk menentukan persyaratan, seperti jumlah karakter minimum, atau apakah kata sandi harus terdiri dari karakter non-alfabet atau tidak.

- Buat kebijakan IAM untuk memberlakukan masuk dengan MFA sehingga para pengguna diizinkan untuk mengelola kata sandi dan perangkat MFA mereka sendiri.

Sumber daya

Praktik-praktik terbaik terkait:

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

Dokumen terkait:

- Kebijakan Kata Sandi Pusat Identitas AWS IAM
- 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 Pusat Identitas AWS IAM
- Menguasai identitas di setiap lapisan beban kerja

SEC02-BP02 Menggunakan kredensial sementara

Saat melakukan autentikasi jenis apa pun, sebaiknya Anda menggunakan kredensial sementara alih-alih kredensial jangka panjang untuk mengurangi atau menghindari risiko, misalnya seperti risiko pengungkapan, pembagian, dan pencurian kredensial.

Hasil yang diinginkan: Untuk mengurangi adanya risiko kredensial jangka panjang, Anda harus menggunakan kredensial sementara sedapat mungkin untuk identitas manusia dan mesin. Kredensial jangka panjang menimbulkan banyak risiko, seperti eksposur melalui unggahan ke

repositori publik. Dengan menggunakan kredensial sementara, Anda dapat secara signifikan mengurangi risiko penyusupan kredensial.

Anti-pola umum:

- Developer memilih menggunakan kunci akses jangka panjang dari pengguna IAM 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 para 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 pun dalam kasus ini kredensial sementara dapat digunakan.

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

Panduan implementasi

Gunakan kredensial keamanan sementara alih-alih kredensial jangka panjang untuk semua permintaan CLI dan API AWS. Permintaan API dan CLI ke layanan AWS harus, dalam hampir setiap kasus, ditandatangani dengan menggunakan kunci akses AWS. Permintaan ini dapat Anda tandatangani dengan menggunakan kredensial jangka panjang maupun sementara. Satu-satunya waktu di mana Anda harus menggunakan kredensial jangka panjang, juga dikenal sebagai kunci akses jangka panjang, adalah jika Anda menggunakan pengguna IAM atau pengguna root Akun AWS. Ketika Anda berfederasi ke AWS atau mengambil peran IAM melalui metode lain, Anda akan menghasilkan kredensial sementara. Bahkan ketika Anda mengakses AWS Management Console dengan 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 dan Anda hampir dapat melakukan semua tugas dengan 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-langkah implementasi

Identitas manusia

Untuk identitas tenaga kerja seperti karyawan, administrator, developer, operator, dan pelanggan:

- Anda harus mengandalkan penyedia identitas tersentralisasi dan meminta pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS dengan menggunakan kredensial sementara. Federasi untuk para pengguna Anda dapat dilakukan baik dengan melakukan federasi langsung ke setiap Akun AWS atau menggunakan Pusat identitas AWS IAM dan penyedia identitas pilihan Anda. Selain mengurangi penggunaan kredensial jangka panjang, federasi memberikan berbagai manfaat dibandingkan penggunaan pengguna IAM. Para pengguna Anda juga dapat meminta kredensial sementara dari baris perintah untuk federasi langsung atau dengan menggunakan Pusat Identitas IAM. Artinya, ada beberapa kasus penggunaan yang memerlukan kredensial jangka panjang atau pengguna IAM untuk pengguna Anda.

Untuk identitas pihak ketiga:

- Saat memberikan akses kepada pihak ketiga, seperti penyedia perangkat lunak sebagai layanan (SaaS), ke sumber daya yang ada di Akun AWS Anda, Anda dapat menggunakan peran lintas akun dan kebijakan berbasis sumber daya. Selain itu, Anda dapat menggunakan alur kredensial klien grant OAuth 2.0 Amazon Cognito untuk pelanggan atau partner SaaS B2B.

Identitas pengguna yang mengakses sumber daya AWS Anda melalui browser web, aplikasi klien, aplikasi seluler, atau alat baris perintah interaktif:

- Jika Anda perlu memberikan akses bagi aplikasi untuk konsumen atau pelanggan ke sumber daya AWS Anda, Anda dapat menggunakan kumpulan identitas Amazon Cognito atau kumpulan pengguna Amazon Cognito untuk memberikan kredensial sementara untuk mereka. Izin untuk kredensial akan dikonfigurasi melalui peran IAM. Anda juga dapat menentukan peran IAM terpisah dengan izin terbatas untuk pengguna tamu yang tidak terautentikasi.

Identitas mesin

Untuk identitas mesin, Anda mungkin perlu menggunakan kredensial jangka panjang. Dalam kasus ini, Anda seharusnya 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 mengonfigurasi peran eksekusi Lambda untuk memberikan izin layanan untuk melakukan tindakan AWS dengan 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 membuat permintaan 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.

Ada skenario di mana kredensial sementara tidak didukung, yang memerlukan penggunaan kredensial jangka panjang. Dalam situasi ini, lakukan audit dan rotasi kredensial secara berkala dan rotasi kunci akses secara rutin. Untuk kunci akses pengguna IAM yang sangat terbatas, pertimbangkan langkah-langkah keamanan tambahan berikut:

- Berikan izin yang sangat terbatas:
  - Patuhi prinsip hak akses paling rendah (tentukan tindakan, sumber daya, dan kondisi spesifik).
  - Pertimbangkan untuk memberi pengguna IAM hanya operasi AssumeRole untuk satu peran tertentu. Bergantung pada arsitektur on-premise, pendekatan ini membantu mengisolasi dan mengamankan kredensial IAM jangka panjang.
- Batasi sumber jaringan dan alamat IP yang diizinkan dalam kebijakan kepercayaan peran IAM.
- Pantau penggunaan dan atur peringatan untuk izin yang tidak digunakan atau penyalahgunaan (menggunakan filter dan alarm metrik Log AWS CloudWatch).
- Berlakukan batasan izin (kebijakan kontrol layanan (SCP) dan batasan izin saling melengkapi - SCP bersifat umum, sementara batasan izin bersifat terperinci).
- Terapkan proses untuk menyediakan dan menyimpan kredensialnya dengan aman (di vault on-premise).

Beberapa opsi lain untuk skenario yang membutuhkan kredensial jangka panjang meliputi:

- Buat API penyedia token Anda sendiri (menggunakan Amazon API Gateway).
- Untuk skenario saat Anda harus menggunakan kredensial jangka panjang, atau kredensial selain kunci akses AWS, seperti login basis data, Anda dapat menggunakan layanan yang dirancang untuk menangani pengelolaan rahasia, seperti AWS Secrets Manager. Secrets Manager menyederhanakan pengelolaan, rotasi, dan penyimpanan rahasia terenkripsi yang aman. Banyak layanan AWS mendukung integrasi langsung dengan Secrets Manager.
- Untuk integrasi multi-cloud, Anda dapat menggunakan federasi identitas berdasarkan kredensial penyedia layanan kredensial (CSP) sumber Anda (lihat AWS STS AssumeRoleWithWebIdentity).

Untuk informasi selengkapnya tentang cara merotasi kredensial jangka panjang, silakan lihat merotasi kunci akses.

Sumber daya

Praktik-praktik terbaik terkait:

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

Dokumen terkait:

- Kredensial Keamanan Sementara
- Kredensial AWS
- Praktik Terbaik Keamanan IAM
- Peran IAM
- Pusat Identitas IAM
- Penyedia Identitas dan Federasi
- Merotasi Kunci Akses
- Solusi Partner Keamanan: Akses dan Kontrol Akses
- Pengguna Root Akun AWS
- Akses AWS menggunakan identitas beban kerja native Google Cloud Platform

- Cara mengakses sumber daya AWS dari penyewa Microsoft Entra ID menggunakan AWS Security Token Service

Video terkait:

- Mengelola izin pengguna dalam skala besar dengan Pusat Identitas IAM AWS
- Menguasai identitas di setiap lapisan beban kerja

SEC02-BP03 Menyimpan dan menggunakan rahasia secara aman

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

Hasil yang diinginkan: Menerapkan sebuah mekanisme untuk mengelola kredensial aplikasi dengan aman yang akan mencapai tujuan-tujuan berikut:

- Melakukan identifikasi atas rahasia apa yang diperlukan untuk beban kerja.
- Mengurangi jumlah kredensial jangka panjang yang diperlukan dan menggunakan kredensial jangka pendek, jika memungkinkan, sebagai gantinya.
- Menetapkan penyimpanan yang aman dan rotasi otomatis kredensial jangka panjang yang tersisa.
- Melakukan audit terhadap akses ke rahasia yang ada di beban kerja.
- Melakukan pemantauan berkelanjutan untuk memverifikasi bahwa tidak ada rahasia yang disematkan di kode sumber selama proses pengembangan.
- Mengurangi kemungkinan terungkapnya kredensial secara tidak sengaja.

Anti-pola umum:

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

Manfaat menjalankan praktik terbaik ini:

- Rahasia yang disimpan dengan enkripsi diam dan bergerak.
- Akses ke kredensial terjaga keamanannya melalui API (bayangkan ia sebagai sebuah mesin penjual otomatis kredensial).
- Akses ke kredensial (baca dan tulis) diaudit dan dicatat log-nya.
- Pemisahan masalah: rotasi kredensial dilakukan oleh komponen terpisah, yang dapat dipisahkan dari bagian arsitektur lainnya.
- Rahasia didistribusikan secara otomatis ke komponen-komponen perangkat lunak sesuai permintaan dan rotasi dilakukan di sebuah lokasi pusat.
- Akses ke kredensial dapat dikontrol dengan sangat ketat dan terperinci.

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

Panduan implementasi

Dahulu, kredensial digunakan untuk melakukan autentikasi ke 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 yang bisa Anda lakukan 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 Anda gunakan untuk menjalankan beban kerja, maka Anda dapat menghapusnya.

Apabila kredensial masih diperlukan untuk menjalankan beban kerja dengan benar, maka kredensial jangka panjangnya mungkin bisa diganti dengan kredensial sementara atau kredensial 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-rahasia ini dapat disimpan dalam sebuah layanan seperti AWS Secrets Manager, di mana rahasia-rahasia tersebut dapat disimpan secara terpusat, dikelola, dan dirotasi secara teratur.

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

Tipe kredensial	Deskripsi	Strategi yang disarankan
Kunci akses IAM	Akses IAM AWS dan kunci rahasia digunakan untuk mengambil peran IAM di dalam sebuah beban kerja	Ganti: Gunakan peran IAM yang ditetapkan ke instans komputasi (seperti Amazon EC2 atau AWS Lambda) sebagai gantinya. Untuk interoperabilitas dengan pihak ketiga yang memerlukan akses ke sumber daya yang ada di Akun AWS Anda, tanyakan apakah mereka mendukung akses lintas akun AWS. Untuk aplikasi seluler, sebaiknya Anda menggunakan kredensial sementara melalui kumpulan identitas Amazon Cognito (identitas terfederasi). Untuk beban kerja yang berjalan di luar AWS, sebaiknya Anda menggunakan IAM Roles Anywhere atau AWS Systems Manager Hybrid Activations. Untuk kontainer, lihat peran IAM tugas Amazon ECS atau peran IAM simpul Amazon EKS.
Kunci SSH	Amankan kunci privat Shell yang digunakan untuk masuk log in ke instans EC2 Linux, secara manual atau sebagai bagian dari proses otomatis	Ganti: Gunakan AWS Systems Manager atau EC2 Instance Connect untuk menyediak an akses terprogram dan manusia ke instans EC2 dengan menggunakan peran IAM.

Tipe kredensial	Deskripsi	Strategi yang disarankan
Kredensial aplikasi dan basis data	Kata sandi – string teks biasa	Rotasi: Simpan kredensial di AWS Secrets Manager dan buat rotasi otomatis jika memungkinkan.
Kredensial Basis Data Admin Amazon RDS dan Aurora	Kata sandi – string teks biasa	Ganti: Gunakan integrasi Secrets Manager dengan Amazon RDS atau Amazon Aurora. Selain itu, beberapa jenis basis data RDS dapat menggunakan peran IAM alih-alih kata sandi untuk beberapa kasus penggunaan (untuk detail lebih lanjut, silakan lihat autentikasi basis data IAM).
Token OAuth	Token rahasia – string teks biasa	Rotasi: Simpan token di AWS Secrets Manager dan konfigurasikan rotasi otomatis.
Token dan kunci API	Token rahasia – string teks biasa	Rotasi: Simpan token di AWS Secrets Manager dan buat rotasi otomatis jika memungkinkan.

Anti-pola umum yang biasa terjadi adalah menyematkan kunci akses IAM ke dalam kode sumber, file konfigurasi, atau aplikasi seluler. Ketika kunci akses IAM diperlukan untuk berkomunikasi dengan layanan AWS, Anda harus menggunakan kredensial keamanan sementara (jangka pendek). Kredensial jangka pendek ini dapat Anda sediakan 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 berinteraksi dengan pihak ketiga, sebaiknya Anda mendelegasikan akses ke peran IAM dengan akses yang diperlukan ke sumber daya akun Anda daripada mengonfigurasi pengguna IAM dan mengirimkan kunci akses rahasia kepada pihak ketiga untuk pengguna tersebut.

Ada banyak kasus di mana beban kerja membutuhkan penyimpanan rahasia yang diperlukan untuk melakukan interoperasi dengan layanan-layanan dan sumber daya lainnya. AWS Secrets Manager dibuat secara khusus untuk mengelola kredensial ini dengan aman, serta mengelola penyimpanan, penggunaan, dan rotasi token API, kata sandi, dan kredensial lainnya.

AWS Secrets Manager menyediakan lima kemampuan utama untuk memastikan penyimpanan yang aman dan penanganan kredensial sensitif: enkripsi saat diam, enkripsi saat bergerak, audit komprehensif, kontrol akses memdetail, dan rotasi kredensial yang dapat diperluas. Anda dapat menggunakan layanan manajemen rahasia lainnya dari Partner AWS atau solusi-solusi yang dikembangkan secara lokal yang dapat memberikan kemampuan dan jaminan serupa.

Saat Anda mengambil rahasia, Anda dapat menggunakan komponen caching sisi klien Secrets Manager untuk men-cache rahasia tersebut untuk digunakan di masa mendatang. Mengambil rahasia yang di-cache lebih cepat daripada mengambilnya dari Secrets Manager. Selain itu, karena ada biaya untuk memanggil API Secrets Manager, penggunaan cache dapat mengurangi biaya Anda. Untuk mengetahui semua cara yang dapat Anda gunakan dalam mengambil rahasia, lihat Dapatkan rahasia.

> **Note**
>
> Beberapa bahasa mungkin mengharuskan Anda menerapkan enkripsi dalam memori Anda sendiri untuk caching sisi klien.

Langkah-langkah implementasi

1. Identifikasi jalur kode yang memuat kredensial yang sudah di-hard-coding dengan menggunakan alat-alat otomatis seperti Amazon CodeGuru..
   a. Gunakan Amazon CodeGuru untuk melakukan pemindaian terhadap repositori kode Anda. Setelah peninjauan selesai dilakukan, lakukan filter pada 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, dan kemudian 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 sebagian dari beban kerja Anda berada di luar AWS tetapi memerlukan kredensial IAM untuk mengakses sumber daya AWS, pertimbangkan 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 runtime.
   a. Konsol CodeGuru dapat secara otomatis membuat sebuah rahasia di Secrets Manager dengan menggunakan kredensial yang ditemukan.
   b. Integrasikan pengambilan rahasia dari Secrets Manager ke dalam kode aplikasi Anda.
      i. Fungsi Lambda nirserver dapat menggunakan ekstensi Lambda yang bersifat agnostik bahasa.
      ii. Untuk instans atau kontainer EC2, AWS memberikan contoh kode sisi klien untuk mengambil rahasia dari Secrets Manager menggunakan beberapa bahasa pemrograman populer.
4. Lakukan peninjauan terhadap basis kode Anda secara berkala dan lakukan pemindaian kembali untuk memverifikasi bahwa tidak ada rahasia baru yang ditambahkan ke kode.
   a. Pertimbangkan untuk menggunakan alat-alat seperti git-secret untuk mencegah memberikan rahasia baru ke repositori kode sumber Anda.
5. Pantau aktivitas Secrets Manager untuk mengetahui adanya indikasi penggunaan yang tidak terduga, akses rahasia yang tidak semestinya, atau upaya untuk menghapus 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-praktik terbaik terkait:

- SEC02-BP02 Menggunakan kredensial sementara
- SEC02-BP05 Melakukan audit dan rotasi kredensial secara berkala

Dokumen terkait:

- Memulai dengan AWS Secrets Manager
- Penyedia Identitas dan Federasi
- Amazon CodeGuru Memperkenalkan Detektor Rahasia
- Cara 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 Secret dalam Skala Besar
- Temukan Rahasia yang Di-Hard-Coding dengan Menggunakan Amazon CodeGuru Secrets Detector
- Mengamankan Rahasia untuk Beban Kerja Hibrida dengan 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 tersentralisasi

Untuk identitas tenaga kerja (karyawan dan kontraktor), andalkan penyedia identitas yang memungkinkan Anda mengelola identitas di sebuah tempat yang tersentralisasi. Hal ini akan mempermudah Anda dalam melakukan 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 multi-faktor (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 (federasi) (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 akan disinkronkan ke penyedia identitas Anda. Misalnya, jika ada seseorang yang keluar dari organisasi Anda, maka Anda dapat secara otomatis mencabut akses ke aplikasi dan sistem gabungan (termasuk AWS) yang dimiliki orang tersebut. Anda telah mengaktifkan pencatatan log audit mendetail di penyedia identitas Anda dan memantau log tersebut untuk mendeteksi perilaku pengguna yang tidak biasa.

Anti-pola 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 melakukan otomatisasi 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 harus 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. Dengan melakukan integrasi dengan sistem sumber daya manusia (SDM), ketika ada seorang pengguna beralih jabatan, perubahan ini akan disinkronkan dengan penyedia identitas yang secara otomatis memperbarui aplikasi dan izin yang ditetapkan. Ketika ada pengguna yang keluar dari organisasi Anda, maka identitas mereka pun secara otomatis dinonaktifkan di penyedia identitas, sehingga akses mereka ke aplikasi dan sistem gabungan dicabut.

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

Panduan implementasi

Panduan untuk pengguna tenaga kerja yang mengakses AWS: Pengguna tenaga kerja seperti karyawan dan kontraktor yang ada di organisasi Anda mungkin memerlukan akses ke AWS dengan 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 melakukan federasi ke beberapa akun di organisasi AWS Anda.

Untuk menggabungkan pengguna tenaga kerja Anda secara langsung dengan masing-masing Akun AWS, Anda dapat menggunakan penyedia identitas terpusat untuk digabungkan ke AWS Identity and Access Management yang ada di akun tersebut. Fleksibilitas IAM dapat memungkinkan Anda mengaktifkan SAMP 2.0 atau Penyedia Identitas Open ID Connect (OIDC) terpisah untuk setiap Akun AWS dan menggunakan atribut-atribut pengguna gabungan untuk kontrol akses. Pengguna tenaga kerja Anda akan menggunakan browser webnya 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 menggunakan 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 SAML dari penyedia identitas.

Untuk melakukan federasi terhadap pengguna tenaga kerja Anda dengan beberapa akun di organisasi AWS Anda, Anda dapat menggunakan Pusat Identitas AWS IAM 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. Pusat Identitas IAM menyediakan direktori sumber identitas default yang dapat Anda gunakan untuk mengelola pengguna dan grup Anda. Atau, Anda dapat memilih sumber identitas eksternal dengan menghubungkan ke penyedia identitas eksternal Anda menggunakan SAML 2.0 dan secara otomatis menyediakan pengguna dan grup menggunakan SCIM, atau menghubungkan ke Microsoft AD Directory Anda dengan menggunakan AWS Directory Service. Setelah Anda mengonfigurasi sumber identitas, Anda dapat menetapkan akses kepada pengguna dan grup ke Akun AWS dengan menentukan kebijakan hak akses paling rendah di rangkaian izin Anda. Pengguna tenaga kerja Anda dapat melakukan autentikasi melalui penyedia identitas pusat Anda untuk masuk ke portal akses AWS dan melakukan masuk tunggal ke aplikasi cloud Akun AWS dan yang ditetapkan untuknya. Pengguna Anda dapat mengonfigurasi AWS CLI v2 untuk melakukan autentikasi dengan Pusat Identitas dan mendapatkan kredensial untuk menjalankan perintah AWS CLI. Pusat Identitas juga dapat memungkinkan akses masuk tunggal ke aplikasi AWS seperti Amazon SageMaker AI Studio dan portal AWS IoT Sitewise Monitor.

Setelah Anda mengikuti panduan di atas, pengguna tenaga kerja Anda tidak perlu lagi menggunakan pengguna IAM 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 sebuah identitas gabungan. Identitas gabungan (terfederasi) menggunakan grup yang ditentukan oleh penyedia identitas terpusat Anda. Anda harus mengidentifikasi dan menghapus grup IAM, pengguna IAM, dan kredensial pengguna jangka panjang (kata sandi dan kunci akses) yang sudah tidak lagi diperlukan di Akun AWS Anda. Anda dapat menemukan kredensial yang tidak digunakan dengan menggunakan laporan kredensial IAM, menghapus pengguna IAM yang sesuai dan menghapus grup IAM. Anda dapat menerapkan Kebijakan Kontrol Layanan (SCP) ke organisasi Anda yang akan membantu Anda dalam mencegah pembuatan pengguna dan grup IAM baru, sehingga memberlakukan bahwa akses ke AWS harus melalui identitas terfederasi.

> **Note**
>
> Anda bertanggung jawab untuk menangani rotasi token akses SCIM seperti yang dijelaskan dalam dokumentasi Penyediaan otomatis. Selain itu, Anda bertanggung jawab untuk merotasi sertifikat yang mendukung federasi identitas Anda.

Panduan untuk para pengguna aplikasi Anda: Anda dapat mengelola identitas pengguna aplikasi Anda, seperti aplikasi seluler, menggunakan Amazon Cognito sebagai penyedia identitas tersentralisasi Anda. Amazon Cognito memungkinkan autentikasi, otorisasi, dan pengelolaan pengguna, baik untuk aplikasi web maupun aplikasi seluler Anda. Amazon Cognito menyediakan toko identitas yang menskalakan jutaan pengguna, mendukung federasi identitas sosial dan organisasi, serta menawarkan fitur-fitur keamanan canggih untuk membantu Anda melindungi para 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-langkah implementasi

Langkah-langkah untuk pengguna tenaga kerja yang mengakses AWS

- Lakukan penggabungan (federasi) terhadap pengguna tenaga kerja Anda ke AWS dengan menggunakan penyedia identitas terpusat melalui salah satu pendekatan berikut:
  - Gunakan Pusat Identitas IAM 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 pengguna IAM 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-praktik terbaik 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 administrasi terdelegasi Pusat Identitas IAM
- Cara menggunakan kebijakan yang dikelola pelanggan di Pusat Identitas IAM untuk kasus penggunaan lanjutan
- AWS CLI v2: Penyedia kredensial Pusat Identitas IAM

Video terkait:

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

Contoh terkait:

- Lokakarya: Menggunakan Pusat Identitas AWS IAM untuk mencapai manajemen identitas yang kuat

Alat terkait:

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

SEC02-BP05 Melakukan audit dan rotasi kredensial secara berkala

Lakukan audit dan rotasi kredensial secara rutin membatasi seberapa lama kredensial dapat digunakan untuk mengakses sumber daya Anda. Kredensial jangka panjang dapat menimbulkan banyak risiko, tetapi risiko-risiko ini dapat dikurangi dengan secara rutin melakukan rotasi terhadap kredensial jangka panjang.

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

Anti-pola umum:

- Tidak melakukan audit penggunaan kredensial.
- Menggunakan kredensial jangka panjang saat tidak diperlukan.
- Menggunakan kredensial jangka panjang dan tidak melakukan rotasi kredensial secara rutin.

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

Panduan implementasi

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

Validasi berkala, sebaiknya menggunakan sebuah alat otomatis, diperlukan untuk memverifikasi bahwa kontrol yang tepat sudah diberlakukan. Untuk identitas manusia, Anda harus mewajibkan pengguna untuk mengubah kata sandi mereka secara rutin dan memensiunkan kunci akses yang ditukar dengan kredensial sementara. Saat Anda berpindah dari pengguna (IAM) AWS Identity and Access Management ke identitas tersentralisasi, Anda dapat membuat laporan kredensial untuk mengaudit pengguna Anda.

Anda juga sebaiknya menerapkan dan memantau MFA dalam penyedia identitas Anda. Anda dapat mengatur Aturan AWS Config, atau menggunakan Standar Keamanan AWS Security Hub,

untuk memantau apakah para pengguna telah mengonfigurasi MFA atau tidak. 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-langkah implementasi

- Melakukan audit kredensial secara rutin: Melakukan audit terhadap identitas yang dikonfigurasi di penyedia identitas Anda dan IAM akan membantu Anda memverifikasi bahwa hanya identitas yang diotorisasi yang memiliki akses ke beban kerja Anda. Identitas tersebut mencakup, tetapi tidak terbatas pada, pengguna IAM, pengguna Pusat Identitas AWS IAM, pengguna Active Directory, atau pengguna dalam penyedia identitas hulu yang berbeda. Misalnya, hapus orang-orang yang keluar dari organisasi, dan hapus pula peran lintas akun yang tidak lagi diperlukan. Sediakan sebuah proses yang dilakukan untuk mengaudit izin ke layanan yang diakses oleh entitas IAM secara berkala. Tindakan ini akan membantu Anda mengidentifikasi kebijakan-kebijakan yang perlu diubah untuk menghapus izin-izin yang tidak digunakan. Gunakan laporan kredensial dan AWS Identity and Access Management Access Analyzer untuk melakukan audit kredensial dan izin IAM. Anda dapat menggunakan Amazon CloudWatch untuk menyiapkan alarm untuk panggilan API tertentu yang dipanggil dalam lingkungan AWS Anda. Amazon GuardDuty juga dapat memperingatkan Anda tentang adanya aktivitas tak terduga, yang mungkin menunjukkan akses yang terlalu permisif atau akses yang tidak diinginkan ke kredensial IAM.
- Rotasi kredensial secara rutin: Ketika Anda tidak dapat menggunakan kredensial sementara, lakukan rotasi terhadap kunci akses IAM jangka panjang secara teratur (maksimum setiap 90 hari). Tindakan ini akan membatasi waktu penggunaan kredensial untuk mengakses sumber daya Anda jika ada kunci akses yang bocor tanpa sepengetahuan Anda. Untuk informasi selengkapnya tentang cara melakukan rotasi terhadap kunci akses untuk pengguna IAM, lihat Merotasi kunci akses.
- Tinjau Izin IAM: Untuk meningkatkan keamanan Akun AWS Anda, lakukan peninjauan dan pemantauan secara rutin terhadap setiap kebijakan IAM Anda. Pastikan bahwa kebijakan tersebut memenuhi prinsip hak akses paling rendah.
- Pertimbangkan untuk mengotomatiskan pembuatan dan pembaruan sumber daya IAM: Pusat Identitas IAM melakukan otomatisasi terhadap banyak tugas IAM, seperti pengelolaan peran dan kebijakan. 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 dapat memungkinkan Anda menggunakan peran di area-area yang biasanya tidak dapat Anda lakukan, 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 memasuki masa kedaluwarsa.

Sumber daya

Praktik-praktik terbaik terkait:

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

Dokumen terkait:

- Memulai dengan 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 Akun AWS Anda

Video terkait:

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

SEC02-BP06 Manfaatkan grup dan atribut pengguna

Penentuan izin sesuai dengan grup pengguna dan atribut akan membantu Anda mengurangi jumlah dan kompleksitas kebijakan, sehingga prinsip hak akses paling rendah dapat Anda wujudkan dengan lebih sederhana. Anda dapat menggunakan grup pengguna untuk mengelola izin bagi banyak orang

di satu tempat berdasarkan fungsi yang mereka lakukan di dalam organisasi Anda. Atribut, seperti departemen, proyek, atau lokasi, dapat menyediakan lapisan tambahan cakupan izin ketika ada orang yang melakukan fungsi serupa, tetapi untuk subset sumber daya yang berbeda.

Hasil yang diinginkan: Anda dapat menerapkan perubahan-perubahan izin berdasarkan fungsi untuk semua pengguna yang menjalankan fungsi tersebut. Keanggotaan grup dan atribut mengatur izin pengguna, sehingga hal itu akan mengurangi kebutuhan untuk mengelola izin di tingkat masing-masing pengguna. Grup dan atribut yang Anda tentukan di penyedia identitas (IdP) Anda disebarkan secara otomatis ke lingkungan AWS Anda.

Anti-pola umum:

- Mengelola izin untuk pengguna individual dan menduplikasinya kepada banyak pengguna.
- Menentukan grup pada tingkat yang terlalu tinggi, sehingga memberikan izin yang terlalu luas.
- Menentukan grup pada tingkat yang terlalu terperinci, sehingga menghasilkan duplikasi dan kebingungan terkait keanggotaan.
- Menggunakan grup dengan izin duplikat di seluruh subset sumber daya ketika atribut dapat digunakan sebagai gantinya.
- Tidak mengelola grup, atribut, dan keanggotaan melalui penyedia identitas standar yang terintegrasi dengan lingkungan AWS Anda.
- Menggunakan rantai peran saat menggunakan sesi Pusat Identitas AWS IAM

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

Panduan implementasi

Izin AWS didefinisikan dalam dokumen yang disebut kebijakan yang dikaitkan dengan principal, seperti pengguna, grup, peran, atau sumber daya. Anda dapat menskalakan manajemen izin dengan mengatur penetapan izin (grup, izin, akun) berdasarkan fungsi pekerjaan, beban kerja, dan lingkungan SDLC. Untuk tenaga kerja Anda, hal ini akan memungkinkan Anda menentukan grup berdasarkan fungsi yang dilakukan pengguna untuk organisasi Anda, bukan berdasarkan sumber daya yang diakses. Misalnya, grup WebAppDeveloper mungkin memiliki kebijakan yang dilampirkan untuk mengonfigurasi layanan seperti Amazon CloudFront dalam akun pengembangan. Grup `AutomationDeveloper` mungkin memiliki beberapa izin yang tumpang-tindih dengan grup `WebAppDeveloper`. Izin ini dapat direkam dalam kebijakan terpisah dan dikaitkan dengan kedua grup, daripada memasukkan pengguna dari kedua fungsi ke dalam sebuah grup `CloudFrontAccess`.

Selain grup, Anda dapat menggunakan atribut untuk menentukan cakupan akses lebih lanjut. Misalnya, Anda mungkin memiliki atribut Proyek untuk pengguna yang ada di grup `WebAppDeveloper` Anda guna menentukan cakupan akses ke sumber daya khusus untuk proyek mereka. Dengan teknik ini, tidak diperlukan grup yang berbeda untuk developer aplikasi yang bekerja di proyek yang berbeda jika izin mereka sama. Cara Anda merujuk ke atribut-atribut yang ada dalam kebijakan izin adalah berdasarkan pada sumbernya, apakah atribut tersebut ditentukan sebagai bagian dari protokol federasi Anda (seperti SAML, OIDC, atau SCIM), sebagai pernyataan SAML kustom, atau ditetapkan dalam Pusat Identitas IAM.

Langkah-langkah implementasi

1. Tetapkan di mana Anda akan menentukan grup dan atribut:
   a. Dengan mengikuti panduan yang diuraikan di SEC02-BP04 Mengandalkan penyedia identitas tersentralisasi, Anda dapat mengetahui apakah Anda perlu menentukan grup dan atribut dalam penyedia identitas Anda, dalam Pusat Identitas IAM, atau menggunakan grup pengguna IAM di akun tertentu.
2. Tentukan grup:
   a. Tentukan grup Anda berdasarkan fungsi dan cakupan akses yang diperlukan. Pertimbangkan untuk menggunakan struktur hierarkis atau konvensi penamaan untuk menyusun grup secara efektif.
   b. Jika Anda menentukannya di dalam Pusat Identitas IAM, maka Anda harus membuat grup dan mengaitkan tingkat akses yang diinginkan dengan menggunakan kumpulan izin.
   c. Jika Anda menentukannya di dalam penyedia identitas eksternal, maka Anda harus menentukan apakah penyedia tersebut mendukung protokol SCIM dan Anda disarankan untuk mengaktifkan penyediaan otomatis dalam Pusat Identitas IAM. Kemampuan ini akan menyinkronkan pembuatan, keanggotaan, dan penghapusan grup antara penyedia Anda dan Pusat Identitas IAM.
3. Tentukan atribut:
   a. Jika Anda menggunakan penyedia identitas eksternal, baik protokol SCIM maupun SAML 2.0 akan menyediakan atribut tertentu secara default. Atribut tambahan dapat didefinisikan dan diteruskan menggunakan pernyataan SAML dengan nama atribut `https://aws.amazon.com/SAML/Attributes/PrincipalTag`. Baca dokumentasi penyedia identitas Anda untuk mengetahui panduan dalam menentukan dan mengonfigurasi atribut kustom.

b. Jika Anda menentukan peran dalam Pusat Identitas IAM, aktifkan fitur kontrol akses berbasis atribut (ABAC) dan tentukan atribut sesuai keinginan. Pertimbangkan atribut yang selaras dengan struktur organisasi atau strategi penandaan sumber daya.

Jika Anda memerlukan rantai peran IAM dari Peran IAM yang diambil melalui Pusat Identitas IAM, nilai seperti `source-identity` dan `principal-tags` tidak akan disebarkan. Untuk detail selengkapnya, lihat Aktifkan dan konfigurasi atribut untuk kontrol akses.

1. Tentukan cakupan izin berdasarkan grup dan atribut:
   a. Sebaiknya Anda menyertakan kondisi dalam kebijakan izin Anda yang membandingkan atribut principal Anda dengan atribut sumber daya yang diakses. Misalnya, Anda dapat menentukan kondisi untuk mengizinkan akses ke sumber daya hanya jika nilai kunci kondisi `PrincipalTag` cocok dengan nilai kunci `ResourceTag` yang memiliki nama yang sama.
   b. Saat mendefinisikan kebijakan ABAC, ikuti panduan dalam praktik terbaik dan contoh otorisasi ABAC.
   c. Tinjau dan perbarui struktur grup dan Anda atribut secara berkala seiring dengan berkembangnya kebutuhan organisasi untuk memastikan pengelolaan izin yang optimal.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC02-BP04 Mengandalkan penyedia identitas tersentralisasi
- SEC03-BP02 Memberikan hak akses paling rendah
- COST02-BP04 Mengimplementasikan grup dan peran

Dokumen terkait:

- Praktik Terbaik IAM
- Kelola Identitas di Pusat Identitas IAM
- Apa itu ABAC untuk AWS?
- ABAC di Pusat Identitas IAM
- Contoh Kebijakan ABAC

Video terkait:

- Mengelola izin pengguna dalam skala besar dengan Pusat Identitas IAM AWS
- Menguasai identitas di setiap lapisan beban kerja

Manajemen izin

Kelola izin guna mengontrol akses untuk identitas orang dan mesin yang memerlukan akses ke AWS dan beban kerja Anda. Izin memungkinkan Anda mengontrol siapa yang dapat mengakses hal tertentu, beserta kondisinya. Dengan menetapkan izin ke identitas manusia dan mesin tertentu, Anda memberinya akses ke tindakan layanan tertentu di sumber daya tertentu. Selain itu, Anda menentukan kondisi yang harus dipenuhi agar akses dapat diberikan.

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 inline, dan dilampirkan ke identitas IAM, termasuk pengguna, grup, atau peran. Kebijakan ini memungkinkan Anda menentukan apa yang dapat dilakukan oleh identitas (izinnya). Kebijakan berbasis identitas dapat dikategorikan lebih lanjut.

Kebijakan terkelola – Kebijakan berbasis identitas mandiri yang dapat diterapkan ke beberapa pengguna, grup, dan peran dalam akun AWS Anda. Ada dua jenis kebijakan terkelola:

- Kebijakan yang dikelola AWS – Kebijakan terkelola yang dibuat dan dikelola oleh AWS.
- Kebijakan yang dikelola pelanggan – Kebijakan terkelola yang Anda buat dan kelola dalam 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 inline mempertahankan hubungan satu-ke-satu yang ketat antara kebijakan dan sebuah 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 pada sumber daya. Sebagai contoh, kebijakan bucket S3 merupakan kebijakan berbasis sumber daya. Kebijakan ini memberikan izin kepada principal

yang dapat berada di akun yang sama atau berbeda dengan sumber daya. Untuk daftar layanan yang mendukung kebijakan berbasis sumber daya, silakan lihat layanan-layanan AWS yang bisa 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): di AWS memungkinkan Anda memberikan izin berdasarkan atribut yang disebut tanda. Tanda dapat dilampirkan pada principal IAM (pengguna atau peran) dan pada sumber daya AWS. Administrator dapat membuat kebijakan IAM yang dapat digunakan kembali yang menerapkan izin berdasarkan atribut principal IAM. Sebagai contoh, sebagai administrator, Anda dapat menggunakan kebijakan IAM tunggal untuk memberi developer di organisasi Anda akses ke sumber daya AWS yang cocok dengan tanda proyek mereka. Seiring tim developer menambahkan sumber daya ke proyek, izin diterapkan secara otomatis berdasarkan atribut, sehingga tidak memerlukan pembaruan kebijakan untuk setiap sumber daya baru.

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

Kebijakan sesi mengambil 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 Tentukan pagar pembatas izin untuk organisasi Anda
- SEC03-BP06 Mengelola akses berdasarkan siklus hidup
- SEC03-BP07 Menganalisis akses publik dan lintas akun

- 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-komponen lainnya. Penetapan harus jelas tentang siapa atau apa yang harus memiliki akses ke tiap-tiap komponen, pilih tipe identitas dan metode autentikasi serta otorisasi yang sesuai.

Anti-pola umum:

- Melakukan hard-coding atau menyimpan rahasia di dalam aplikasi Anda.
- Memberikan izin kustom untuk masing-masing pengguna.
- Menggunakan kredensial yang sudah berumur panjang.

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

Panduan implementasi

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

Akses reguler ke Akun AWS dalam organisasi harus disediakan dengan menggunakan akses federasi atau penyedia identitas tersentralisasi. Anda juga sebaiknya melakukan sentralisasi manajemen identitas Anda dan memastikan terdapat praktik yang mapan yang digunakan untuk mengintegrasikan akses AWS ke siklus hidup akses karyawan Anda. Misalnya, saat ada seorang karyawan yang berganti peran pekerjaan dengan level akses berbeda, maka keanggotaan grupnya juga harus berubah agar sesuai dengan persyaratan akses baru yang berlaku padanya.

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

Layanan-layanan AWS, seperti AWS Secrets Manager dan Systems Manager Parameter Store AWS, dapat membantu Anda memisahkan rahasia dari aplikasi atau beban kerja dengan aman jika tidak layak untuk menggunakan peran IAM. 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 IAM 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 peran IAM yang sama yang Anda gunakan dengan aplikasi AWS untuk mengakses sumber daya AWS.

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

Pengguna membutuhkan akses terprogram jika ingin berinteraksi dengan AWS di luar AWS Management Console. Cara memberikan akses programatis bergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensial sementara untuk menandatangani permintaan programatis ke AWS CLI, SDK AWS, atau API AWS.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengonfigurasi AWS CLI untuk menggunakan AWS IAM Identity Center dalam Panduan Pengguna AWS Command Line Interface. • Untuk SDK AWS, alat, dan API AWS, lihat Autentikasi

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		Pusat Identitas IAM dalam Panduan Referensi SDK dan Alat AWS.
IAM	Gunakan kredensial sementara untuk menandatangani permintaan programatis ke AWS CLI, SDK AWS, atau API AWS.	Ikuti petunjuk dalam Menggunakan kredensial sementara dengan sumber daya AWS dalam Panduan Pengguna IAM.
IAM	(Tidak direkomendasikan) Gunakan kredensial jangka panjang untuk menandatangani permintaan programatis ke AWS CLI, AWS SDK, atau API AWS.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengautentikasi menggunakan kredensial pengguna IAM dalam Panduan Pengguna AWS Command Line Interface. • Untuk SDK dan alat AWS, lihat Mengautentikasi menggunakan kredensial jangka panjang dalam Panduan Referensi SDK dan Alat AWS. • Untuk API AWS, lihat Mengelola kunci akses untuk pengguna IAM dalam Panduan Pengguna IAM.

Sumber daya

Dokumen terkait:

- Kontrol akses berbasis atribut (ABAC)
- AWS IAM Identity Center
- IAM Roles Anywhere
- Kebijakan terkelola AWS untuk Pusat Identitas IAM
- Ketentuan kebijakan IAM AWS
- Kasus penggunaan IAM
- Hapus kredensial yang tidak perlu
- Bekerja dengan Kebijakan
- Cara mengontrol akses ke sumber daya AWS berdasarkan Akun AWS, OU, atau organisasi
- Identifikasi, atur, dan kelola rahasia secara mudah dengan 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

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

Hasil yang diinginkan: Pengguna hanya memiliki izin minimum yang diperlukan untuk fungsi pekerjaan spesifik mereka. Anda menggunakan Akun AWS terpisah untuk mengisolasi developer dari lingkungan produksi. Ketika developer perlu mengakses lingkungan produksi untuk tugas-tugas tertentu, mereka diberi akses terbatas dan terkontrol hanya selama durasi tugas tersebut. Akses

produksi mereka segera dicabut setelah mereka menyelesaikan pekerjaan yang diperlukan. Anda melakukan peninjauan reguler atas izin dan segera mencabutnya saat tidak diperlukan lagi, seperti saat pengguna berganti peran atau meninggalkan organisasi. Anda membatasi hak administrator ke grup kecil yang tepercaya untuk mengurangi paparan risiko. Anda memberi akun mesin atau sistem hanya izin minimum yang diperlukan untuk melakukan tugas yang dimaksudkan.

Anti-pola umum:

- Secara default, Anda memberikan izin administrator kepada pengguna.
- Anda menggunakan akun pengguna root untuk aktivitas sehari-hari.
- Anda membuat kebijakan yang terlalu permisif tanpa cakupan yang tepat.
- Peninjauan izin Anda jarang dilakukan, sehingga menyebabkan penyimpangan izin.
- Anda hanya mengandalkan kontrol akses berbasis atribut untuk isolasi lingkungan atau manajemen izin.

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

Panduan implementasi

Prinsip hak akses paling rendah menyatakan bahwa identitas hanya boleh mendapatkan izin untuk melakukan serangkaian tindakan terkecil yang diperlukan untuk memenuhi tugas tertentu. Hal ini akan menyeimbangkan kegunaan, efisiensi, dan keamanan. Pengoperasian berdasarkan prinsip ini akan membantu Anda membatasi akses yang tidak diinginkan dan membantu Anda dalam memantau siapa saja yang memiliki akses ke sumber daya yang mana. Pengguna IAM dan peran IAM tidak memiliki izin secara default. Pengguna root memiliki akses penuh secara default dan harus dikontrol, dipantau, dan digunakan secara ketat hanya untuk tugas-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 Anda 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. Misalnya, 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 dapat 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 beban kerja Anda membuat kebijakan IAM mereka sendiri untuk sistem yang mereka bangun, tetapi hanya jika mereka menerapkan Batasan Izin untuk membatasi izin maksimum yang dapat mereka berikan.

Langkah-langkah implementasi

- Implementasikan kebijakan hak akses paling rendah: Tetapkan kebijakan akses dengan hak akses paling rendah ke grup dan peran IAM untuk mencerminkan peran atau fungsi pengguna yang telah Anda tetapkan.
- Isolasikan lingkungan pengembangan dan produksi melalui Akun AWS terpisah: Gunakan Akun AWS terpisah untuk lingkungan pengembangan dan produksi, serta kontrol akses di antara keduanya menggunakan kebijakan kontrol layanan, kebijakan sumber daya, dan kebijakan identitas.
- Kebijakan dasar penggunaan API: Salah satu cara untuk menentukan izin yang diperlukan adalah dengan melakukan peninjauan terhadap log AWS CloudTrail. Anda dapat menggunakan peninjauan ini untuk membuat izin yang disesuaikan dengan tindakan yang benar-benar dilakukan oleh pengguna di dalam AWS. IAM Access Analyzer dapat secara otomatis menghasilkan sebuah kebijakan IAM berdasarkan aktivitas akses. Anda dapat menggunakan IAM Access Advisor di tingkat organisasi atau akun untuk melacak informasi yang terakhir diakses untuk kebijakan tertentu.
- Pertimbangkan untuk menggunakan kebijakan terkelola AWS untuk fungsi pekerjaan: Saat Anda mulai membuat kebijakan izin yang terperinci, sebaiknya gunakan kebijakan terkelola AWS untuk peran pekerjaan umum, seperti penagihan, administrator basis data, dan ilmuwan data. Kebijakan ini dapat membantu Anda mempersempit akses yang dimiliki pengguna sambil menentukan cara menerapkan kebijakan hak akses paling rendah.
- Hapus izin yang tidak perlu: Deteksi dan hapus entitas, kredensial, dan izin IAM yang tidak digunakan untuk mewujudkan prinsip hak akses paling rendah. Anda dapat menggunakan IAM Access Analyzer untuk mengidentifikasi akses eksternal dan tidak terpakai, serta pembuatan kebijakan IAM Access Analyzer dapat membantu menyempurnakan kebijakan izin.
- Pastikan bahwa para pengguna memiliki akses terbatas ke lingkungan produksi: Pengguna hanya boleh memiliki akses ke lingkungan produksi jika memiliki kasus penggunaan yang valid. Setelah

pengguna menyelesaikan tugas-tugas tertentu yang memerlukan akses produksi, akses harus dicabut. Pembatasan akses ke lingkungan produksi akan membantu Anda mencegah kejadian tak terduga yang memengaruhi produksi dan memperkecil cakupan dampak akses yang tidak diharapkan.

- Pertimbangkan batasan izin: Batasan izin adalah sebuah fitur untuk menggunakan sebuah kebijakan terkelola yang mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke sebuah entitas IAM. Batasan izin entitas mengizinkannya untuk melakukan hanya tindakan yang diizinkan oleh kebijakan berbasis identitas dan batasan izinnya.
- Efektifkan akses menggunakan kontrol akses berbasis atribut dan tanda sumber daya: Kontrol akses berbasis atribut (ABAC) yang menggunakan tanda sumber daya dapat digunakan untuk mengefektifkan izin jika didukung. Anda dapat menggunakan model ABAC yang membandingkan tanda principal dengan tanda sumber daya untuk mengefektifkan akses berdasarkan dimensi kustom yang Anda tentukan. Pendekatan ini dapat menyederhanakan dan mengurangi jumlah kebijakan izin di organisasi Anda.
  - Sebaiknya ABAC hanya digunakan untuk kontrol akses ketika principal dan sumber daya dimiliki oleh Organisasi AWS Anda. Pihak eksternal dapat menggunakan nama dan nilai tanda yang sama dengan organisasi Anda untuk principal dan sumber daya mereka sendiri. Jika Anda hanya mengandalkan pasangan nama-nilai ini untuk memberikan akses ke principal atau sumber daya pihak eksternal, Anda mungkin akan memberikan izin yang tidak dimaksudkan.
- Gunakan kebijakan kontrol layanan untuk AWS Organizations: Kebijakan kontrol layanan secara terpusat mengontrol izin maksimum yang tersedia bagi akun anggota yang ada di organisasi Anda. Yang terpenting, Anda dapat menggunakan kebijakan kontrol layanan untuk membatasi izin pengguna root di dalam akun anggota. Pertimbangkan juga untuk menggunakan AWS Control Tower, yang akan menyediakan kontrol terkelola preskriptif yang akan makin memperkaya AWS Organizations. Anda juga dapat menentukan kontrol Anda sendiri di dalam Control Tower.
- Menetapkan sebuah kebijakan siklus hidup pengguna untuk organisasi Anda: Kebijakan siklus hidup pengguna akan menentukan tugas yang akan dilakukan saat pengguna berada di AWS, mengubah peran atau cakupan pekerjaan, atau tidak lagi memerlukan akses ke AWS. Lakukan peninjauan izin pada setiap langkah dalam siklus hidup pengguna untuk memverifikasi bahwa izin dibatasi dengan sesuai dan untuk menghindari penyimpangan izin.
- Tetapkan jadwal reguler untuk meninjau izin dan menghapus izin yang tidak diperlukan: Anda harus secara teratur melakukan peninjauan akses pengguna untuk memverifikasi bahwa pengguna tidak memiliki akses yang terlalu permisif. AWS Config dan IAM Access Analyzer dapat membantu Anda selama audit izin pengguna.

- Tetapkan matriks peran pekerjaan: Matriks peran pekerjaan memberikan visualisasi dari berbagai peran dan tingkat akses yang diperlukan dalam jejak AWS Anda. Dengan sebuah matriks peran kerja, Anda dapat menentukan dan memisahkan izin berdasarkan tanggung jawab pengguna di dalam organisasi. Gunakan grup alih-alih menerapkan izin langsung ke masing-masing pengguna atau peran.

Sumber daya

Dokumen terkait:

- Berikan hak akses paling rendah
- Batasan izin untuk entitas IAM
- Teknik untuk menulis kebijakan IAM dengan hak akses paling rendah
- IAM Access Analyzer akan mempermudah Anda dalam mengimplementasikan izin dengan hak akses paling rendah dengan membuat kebijakan IAM berdasarkan aktivitas akses
- Delegasikan manajemen izin kepada developer dengan menggunakan batasan izin IAM
- Menyempurnakan Izin dengan menggunakan informasi yang terakhir kali diakses
- Tipe 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 Mengatur Lingkungan Anda dan Mendorong Akuntabilitas
- Strategi Penandaan AWS
- Penandaan pada sumber daya AWS

Video terkait:

- Manajemen izin generasi berikutnya
- Zero Trust: Sebuah perspektif AWS

SEC03-BP03 Menerapkan proses akses darurat

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

Anda harus merancang desain proses untuk berbagai mode kegagalan yang dapat mengakibatkan terjadinya sebuah peristiwa darurat. Misalnya, dalam keadaan normal, pengguna tenaga kerja Anda melakukan federasi ke cloud menggunakan penyedia identitas tersentralisasi (SEC02-BP04) untuk mengelola beban kerja mereka. Namun, jika penyedia identitas tersentralisasi Anda gagal, atau konfigurasi untuk federasi di cloud diubah, maka pengguna tenaga kerja Anda mungkin tidak dapat melakukan federasi ke cloud. Proses akses darurat akan 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 tersebut digunakan sampai mekanisme federasi normal berhasil dipulihkan.

Hasil yang diinginkan:

- Anda telah menentukan dan mendokumentasikan mode kegagalan yang terhitung sebagai sebuah keadaan darurat: pertimbangkan keadaan normal Anda dan sistem yang diandalkan oleh para pengguna untuk mengelola beban kerja mereka. Pertimbangkan bagaimana masing-masing dependensi ini dapat gagal dan menyebabkan terjadinya sebuah keadaan darurat. Anda mungkin akan menemukan pertanyaan dan praktik-praktik terbaik dalam Pilar Keandalan yang berguna untuk mengidentifikasi mode kegagalan dan merancang arsitektur sistem yang lebih tangguh untuk meminimalkan kemungkinan terjadinya kegagalan.
- Anda telah mendokumentasikan langkah-langkah yang harus diikuti untuk mengonfirmasi bahwa kegagalan dianggap sebagai keadaan darurat. Misalnya, Anda dapat meminta administrator identitas Anda untuk memeriksa status penyedia identitas utama dan siaga Anda dan, jika keduanya tidak tersedia, maka Anda harus mengumumkan peristiwa darurat untuk kegagalan penyedia identitas.
- Anda telah menentukan sebuah proses akses darurat khusus untuk masing-masing jenis mode darurat atau kegagalan. Pengkhususan ini dapat mengurangi godaan di pihak pengguna Anda untuk terlalu sering menggunakan sebuah proses umum untuk semua jenis keadaan darurat. Proses akses darurat Anda menggambarkan keadaan di mana masing-masing 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 dengan efisien. Ingatlah bahwa sebuah peristiwa darurat dapat menjadi saat-saat yang memusingkan bagi para pengguna Anda dan mereka sedang berada di bawah tekanan waktu yang ekstrem, jadi buatlah desain proses yang sesederhana mungkin.

Anti-pola umum:

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

Manfaat menjalankan praktik terbaik ini:

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

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

Panduan implementasi

Bagian ini akan memberikan Anda panduan untuk 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 yang berlaku.

Panduan umum untuk semua mode kegagalan

Pertimbangkan hal berikut saat Anda membuat desain proses akses darurat untuk sebuah mode kegagalan:

- Dokumentasikan kondisi awal (pre-conditions) dan asumsi mengenai proses tersebut: kapan proses tersebut harus digunakan dan kapan proses tersebut tidak boleh digunakan. Penting untuk memiliki detail mode kegagalan dan mendokumentasikan asumsi, seperti status keadaan sistem-sistem terkait lainnya. Misalnya, proses untuk Mode Kegagalan 2 mengasumsikan bahwa penyedia identitas tersedia, tetapi konfigurasi yang ditetapkan di AWS sudah dimodifikasi atau telah kedaluwarsa.
- Sejak awal, buat sumber daya yang dibutuhkan oleh proses akses darurat (SEC10-BP05). Misalnya, buat akses Akun AWS darurat di awal dengan pengguna IAM dan peran IAM, dan peran IAM lintas akun di semua akun beban kerja. Hal ini akan memastikan bahwa semua sumber daya ini siap dan tersedia ketika ada sebuah peristiwa darurat yang terjadi. Dengan membuat sumber daya sebelumnya, artinya Anda tidak memiliki dependensi pada API bidang kontrol AWS (digunakan untuk membuat dan memodifikasi sumber daya AWS) yang mungkin tidak akan tersedia ketika ada keadaan darurat yang terjadi. Selanjutnya, dengan membuat sumber daya IAM sebelumnya, artinya Anda tidak perlu memperhitungkan potensi penundaan yang disebabkan terjadinya konsistensi di akhir.
- Sertakan proses-proses akses darurat sebagai bagian dari rencana manajemen insiden Anda (SEC10-BP02). Dokumentasikan bagaimana peristiwa darurat dilacak dan dikomunikasikan kepada orang lain yang ada dalam organisasi Anda, seperti tim sejawat, pimpinan Anda, dan, jika ada, secara eksternal kepada para pelanggan dan partner bisnis Anda.
- Tentukan proses permintaan akses darurat yang ada dalam sistem alur kerja permintaan layanan yang ada, jika Anda memilikinya. Biasanya, sistem alur kerja semacam ini akan memungkinkan Anda untuk membuat formulir penerimaan informasi untuk mengumpulkan informasi tentang permintaan, melacak permintaan melalui setiap tahap alur kerja, dan menambahkan langkah-langkah persetujuan otomatis dan manual. Hubungkan setiap permintaan dengan sebuah peristiwa darurat terkait yang dilacak dalam sistem manajemen insiden Anda. Dengan memiliki sistem yang seragam untuk akses darurat, Anda dapat melacak permintaan tersebut dalam satu sistem tunggal, menganalisis tren penggunaan, dan meningkatkan kualitas proses Anda.
- Pastikan bahwa proses akses darurat Anda hanya dapat dimulai oleh pengguna yang berwenang dan itu 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 bagaimana permintaan itu dieskalasikan ke rantai manajemen Anda hingga mendapatkan persetujuan.

- Terapkan mekanisme pencatatan log, pemantauan, dan peringatan yang efektif untuk proses dan mekanisme akses darurat. Buat log audit yang mendetail untuk semua upaya yang berhasil dan upaya yang gagal dalam mendapatkan akses darurat. Korelasikan aktivitas dengan peristiwa darurat yang sedang berlangsung dari sistem manajemen insiden Anda, dan jalankan peringatan jika tindakan terjadi di luar periode waktu yang diharapkan atau jika akun akses darurat digunakan selama operasi normal. Akun akses darurat hanya boleh diakses selama keadaan darurat karena prosedur break-glass dapat dianggap sebagai backdoor. Integrasikan dengan alat informasi keamanan dan manajemen peristiwa (SIEM) Anda atau AWS Security Hub untuk melaporkan dan mengaudit semua aktivitas selama periode akses darurat. Setelah kembali ke operasi normal, rotasikan kredensial akses darurat secara otomatis, dan beri tahu tim yang relevan.
- Lakukan pengujian terhadap proses akses darurat secara berkala untuk memverifikasi bahwa langkah-langkahnya sudah 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 melakukan federasi ke AWS tidak tersedia

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

Apabila penyedia identitas tersentralisasi 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 tersentralisasi 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 terjadinya lonjakan lalu lintas pelanggan yang

tidak terduga. Administrator darurat Anda harus mengikuti proses akses darurat untuk mendapatkan akses ke Akun AWS khusus produksi dan melakukan perubahan-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 dan pengguna IAM) yang digunakan untuk mendukung proses akses darurat tersebut. Selama operasi normal, tidak ada yang boleh mengakses akun akses darurat tersebut dan Anda harus melakukan pemantauan atas dan memberikan peringatan mengenai terjadinya 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 yang ada di Akun AWS yang memerlukan akses darurat. Peran IAM ini telah dibuat sebelumnya dan dikonfigurasi dengan kebijakan-kebijakan kepercayaan yang mempercayai peran IAM yang dimiliki akun darurat tersebut.

Proses akses darurat dapat menggunakan salah satu pendekatan berikut ini:

- Anda dapat membuat satu set pengguna IAM untuk administrator darurat Anda yang ada di akun akses darurat dengan kata sandi yang kuat dan token MFA yang sudah dikaitkan. Set pengguna IAM ini memiliki izin untuk mengambil peran IAM yang kemudian akan memungkinkan akses lintas akun ke Akun AWS tempat di mana akses darurat diperlukan. Kami sarankan Anda untuk membuat pengguna sesedikit mungkin dan menetapkan masing-masing pengguna ke satu administrator darurat. Selama keadaan darurat, pengguna administrator darurat masuk ke akun akses darurat dengan menggunakan kata sandi dan kode token MFA mereka, beralih ke peran IAM akses darurat yang ada di akun darurat, dan pada akhirnya beralih ke peran IAM akses darurat yang ada di akun beban kerja untuk melakukan tindakan perubahan darurat. Kelebihan pendekatan ini adalah setiap pengguna IAM ditugaskan ke satu administrator darurat dan Anda dapat mengetahui pengguna mana yang masuk dengan melakukan peninjauan pada peristiwa CloudTrail. Kelemahan pendekatan ini adalah Anda harus mempertahankan beberapa pengguna IAM dengan kata sandi berumur panjang dan token MFA yang sudah dikaitkan.
- 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 yang ada di akun beban kerja. Kami merekomendasikan Anda untuk menggunakan pengaturan kata sandi yang kuat dan beberapa token MFA untuk pengguna root tersebut. Kami juga menyarankan Anda untuk menyimpan kata sandi dan token MFA di sebuah brankas kredensial korporasi yang aman yang memberlakukan autentikasi dan otorisasi yang kuat. Anda harus mengamankan kata sandi dan faktor pengaturan ulang token MFA: atur alamat email akun ke sebuah daftar distribusi email yang dipantau oleh administrator keamanan cloud Anda, dan atur nomor telepon

akun ke nomor telepon bersama yang juga dipantau oleh administrator keamanan. Keunggulan pendekatan ini adalah bahwa hanya ada satu set kredensial pengguna root yang harus dikelola. Kelemahan pendekatan ini adalah karena ini merupakan pengguna bersama, maka ada beberapa administrator yang memiliki kemampuan untuk masuk sebagai pengguna root. Anda harus melakukan audit terhadap peristiwa log brankas korporasi Anda untuk mengidentifikasi administrator mana yang menggunakan kata sandi pengguna root.

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

Agar pengguna tenaga kerja Anda dapat melakukan federasi ke Akun AWS, Anda dapat mengonfigurasi Pusat Identitas IAM dengan penyedia identitas eksternal atau membuat sebuah Penyedia Identitas IAM (SEC02-BP04). Biasanya, Anda melakukan konfigurasi dengan mengimpor dokumen XML metadata SAML yang disediakan oleh penyedia identitas Anda. Dokumen metadata XML tersebut menyertakan sebuah 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 seorang administrator. Dalam skenario lain, sertifikat X.509 yang diimpor ke dalam AWS dapat kedaluwarsa dan XML metadata yang baru yang memiliki sertifikat baru belum diimpor ke AWS. Kedua skenario ini dapat mengganggu federasi ke AWS untuk pengguna tenaga kerja Anda, dan hal ini akan mengakibatkan terjadinya sebuah keadaan darurat.

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

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

Mode Kegagalan 3: Gangguan Pusat Identitas

Apabila terjadi gangguan Wilayah AWS atau terjadi peristiwa yang tidak semestinya pada Pusat Identitas IAM, kami sarankan Anda untuk 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 yang ada dalam sebuah akun darurat. Untuk mendapatkan detail tentang proses dan pertimbangan desain, silakan lihat Mengatur akses darurat ke AWS Management Console.

Langkah-langkah implementasi

Langkah-langkah umum untuk semua mode kegagalan

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

- Buatlah kebijakan kontrol layanan (SCP) AWS Organizations untuk menyangkal penghapusan dan modifikasi peran IAM lintas akun yang ada dalam anggota Akun AWS.
- Aktifkan CloudTrail untuk Akun AWS akses darurat dan kirimkan peristiwa jejak ke bucket S3 pusat yang ada di Akun AWS pengumpulan log Anda. Jika Anda menggunakan AWS Control Tower untuk menyiapkan dan mengatur lingkungan multi-akun AWS Anda, maka setiap akun yang Anda buat dengan menggunakan AWS Control Tower atau Anda daftarkan di AWS Control Tower akan memiliki CloudTrail yang diaktifkan secara default dan dikirim ke bucket S3 dalam sebuah Akun AWS arsip log khusus.
- Pantau aktivitas yang terjadi 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 ada aktivitas yang terjadi di luar peristiwa darurat yang sedang berlangsung yang terlacak 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 sudah dimodifikasi atau telah kedaluwarsa

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

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

- Sebagaimana diuraikan dalam langkah Siapkan akses darurat ke AWS Management Console, di Akun AWS akses darurat, buat sebuah Penyedia Identitas IAM untuk mengaktifkan federasi SAML langsung dari penyedia identitas Anda.
- Buat grup operasi darurat di IdP Anda tanpa anggota.
- Buat peran IAM yang sesuai dengan grup operasi darurat yang ada di akun akses darurat.

Sumber daya

Praktik terbaik Well-Architected terkait:

- SEC02-BP04 Mengandalkan penyedia identitas tersentralisasi
- 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 gabungan SAML 2.0 untuk mengakses AWS Management Console
- Akses pecah kaca

Video terkait:

- AWS re:invent 2022 - Menyederhanakan akses tenaga kerja Anda dengan Pusat Identitas IAM

- AWS re:Inforce 2022 - Pembahasan mendalam tentang 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, maka Anda harus menghapus izin-izin yang tidak diperlukan dan menetapkan proses peninjauan untuk mendapatkan izin hak akses paling rendah. Lakukan pemantauan secara terus-menerus dan hapus identitas serta izin-izin yang tidak diperlukan, baik untuk akses manusia maupun mesin.

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

Anti-pola umum:

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

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

Panduan implementasi

Setelah tim dan proyek mulai, kebijakan izin permisif mungkin digunakan untuk menumbuhkan banyak inovasi dan ketangkasan. Misalnya, di dalam sebuah lingkungan pengembangan atau pengujian, developer dapat diberi akses ke berbagai 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 sebuah lingkungan produksi, yang apabila izinnya terlalu permisif, maka dampaknya bisa begitu luas dan berpotensi mengekspos data konsumen.

AWS menyediakan berbagai metode untuk membantu Anda mengidentifikasi pengguna, peran, izin, dan kredensial yang tidak diperlukan. AWS juga dapat membantu Anda dalam menganalisis aktivitas akses yang dilakukan oleh pengguna IAM 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-kebijakan pembatasan izin berdasarkan layanan dan tindakan aktual yang berinteraksi dengan principal. Kontrol akses berbasis atribut (ABAC) dapat membantu Anda untuk menyederhanakan pengelolaan izin, karena Anda dapat memberikan izin kepada para pengguna dengan menggunakan atribut mereka, alih-alih melampirkan kebijakan izin secara langsung ke masing-masing pengguna.

Langkah-langkah implementasi

- Gunakan AWS Identity and Access Management Access Analyzer: IAM Access Analyzer akan membantu Anda mengidentifikasi sumber daya yang ada di akun dan organisasi Anda, seperti bucket Amazon Simple Storage Service (Amazon S3) atau peran IAM, yang digunakan bersama dengan sebuah entitas eksternal.
- Gunakan pembuatan kebijakan IAM Access Analyzer: Pembuatan kebijakan IAM Access Analyzer membantu Anda membuat kebijakan izin yang sangat terperinci berdasarkan aktivitas akses pengguna IAM atau peran IAM.
- Uji izin di lingkungan yang lebih rendah sebelum produksi: Mulailah dengan memanfaatkan lingkungan sandbox dan pengembangan yang tidak begitu krusial guna menguji izin yang diperlukan untuk berbagai fungsi pekerjaan menggunakan IAM Access Analyzer. Kemudian, secara bertahap perketat dan validasikan izin ini di seluruh lingkungan pengujian, jaminan kualitas, dan pementasan sebelum menerapkannya pada produksi. Lingkungan yang lebih rendah dapat memiliki izin yang lebih longgar pada awalnya karena kebijakan kontrol layanan (SCP) memberlakukan pagar pembatas dengan membatasi izin maksimum yang diberikan.
- Menentukan jangka waktu dan kebijakan penggunaan yang dapat diterima untuk pengguna IAM dan peran IAM: Gunakan stempel waktu yang terakhir diakses untuk mengidentifikasi pengguna dan peran yang tidak digunakan dan kemudian menghapusnya. Tinjau layanan dan tindakan informasi yang terakhir diakses untuk mengidentifikasi dan masukkan izin dalam cakupan untuk 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-tindakan tersebut. Fitur informasi yang terakhir diakses tersedia di AWS Management Console dan secara terprogram akan memungkinkan Anda untuk menggabungkannya ke dalam alur kerja infrastruktur dan alat-alat otomatis Anda.

- Pertimbangkan untuk mencatat 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 untuk peristiwa ini sehingga Anda bisa menentukan pengguna dan peran apa yang perlu mengakses objek Amazon S3 dan item tabel DynamoDB tertentu.

Sumber daya

Dokumen terkait:

- Berikan hak akses paling rendah
- Hapus kredensial yang tidak perlu
- Apa itu AWS CloudTrail?
- Bekerja dengan Kebijakan
- Pencatatan log dan pemantauan DynamoDB
- Menggunakan pencatatan log peristiwa CloudTrail untuk bucket dan objek Amazon S3
- Mendapatkan laporan kredensial untuk akun 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 - Pembahasan mendalam tentang AWS Identity and Access Management (IAM)

SEC03-BP05 Tentukan pagar pembatas izin untuk organisasi Anda

Gunakan pagar pembatas izin untuk mengurangi cakupan izin-izin yang tersedia yang dapat diberikan kepada principal. Rantai evaluasi kebijakan izin mencakup pagar pembatas yang digunakan untuk menentukan izin efektif principal saat membuat keputusan otorisasi. Anda dapat menentukan pagar pembatas dengan menggunakan pendekatan berbasis lapisan. Terapkan

beberapa pagar pembatas secara meluas di seluruh organisasi Anda dan terapkan pagar pembatas lainnya secara terperinci ke sesi akses sementara.

Hasil yang diinginkan: Anda memiliki isolasi lingkungan yang jelas menggunakan Akun AWS terpisah. Kebijakan kontrol layanan (SCP) digunakan untuk menentukan pagar pembatas izin di tingkat organisasi. Pagar pembatas yang lebih meluas ditetapkan pada tingkat hierarki yang paling dekat dengan root organisasi Anda, dan pagar pembatas yang lebih ketat ditetapkan lebih dekat ke tingkat akun individual.

Jika didukung, kebijakan sumber daya akan menentukan kondisi yang harus dipenuhi oleh principal untuk mendapatkan akses ke sebuah sumber daya. Kebijakan sumber daya juga mengurangi cakupan dari serangkaian tindakan yang diizinkan, jika sesuai. Batasan izin diterapkan pada principal yang mengelola izin beban kerja, dengan mendelegasikan manajemen izin kepada pemilik beban kerja individual.

Anti-pola umum:

- Membuat Akun AWS anggota dalam suatu Organisasi AWS, tetapi tidak menggunakan SCP untuk membatasi penggunaan dan izin yang tersedia untuk kredensial root-nya.
- Menetapkan izin berdasarkan hak akses paling rendah, tetapi tidak menerapkan pagar pembatas pada kumpulan izin maksimum yang dapat diberikan.
- Mengandalkan fondasi penolakan implisit AWS IAM untuk membatasi izin, yang meyakini bahwa kebijakan tidak akan memberikan izin eksplisit yang tidak diinginkan.
- Menjalankan beberapa lingkungan beban kerja dalam lingkungan Akun AWS yang sama, dan kemudian mengandalkan berbagai mekanisme, seperti VPC, tanda, atau kebijakan sumber daya untuk memberlakukan batasan izin.

Manfaat menjalankan praktik terbaik ini: Pagar pembatas izin akan membantu Anda untuk membangun keyakinan bahwa izin yang tidak diinginkan tidak dapat diberikan, bahkan ketika kebijakan izin mencoba melakukannya. Hal ini dapat menyederhanakan penentuan dan pengelolaan izin dengan mengurangi cakupan maksimum izin yang perlu dipertimbangkan.

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

Panduan implementasi

Sebaiknya Anda gunakan pendekatan berbasis lapisan untuk menentukan pagar pembatas izin bagi organisasi Anda. Pendekatan ini secara sistematis akan mengurangi izin maksimum yang mungkin

digunakan saat lapisan tambahan diterapkan. Hal ini akan membantu Anda memberikan akses berdasarkan prinsip hak akses paling rendah, sehingga itu akan mengurangi risiko akses yang tidak diinginkan karena terjadinya kesalahan konfigurasi kebijakan.

Langkah pertama untuk membuat pagar pembatas izin adalah mengisolasi beban kerja dan lingkungan Anda ke dalam Akun AWS terpisah. Principal dari satu akun tidak dapat mengakses sumber daya yang ada di akun lain tanpa izin eksplisit untuk melakukannya, bahkan ketika kedua akun berada di organisasi AWS yang sama atau di bawah unit organisasi (OU) yang sama. Anda dapat menggunakan OU untuk mengelompokkan akun-akun yang ingin Anda kelola sebagai satu unit tunggal.

Langkah selanjutnya adalah mengurangi izin maksimum yang dapat Anda berikan kepada principal yang ada dalam akun anggota di organisasi Anda. Anda dapat menggunakan kebijakan kontrol layanan (SCP) untuk tujuan ini, yang dapat Anda terapkan pada sebuah OU atau akun. SCP dapat memberlakukan kontrol akses umum, seperti membatasi akses ke Wilayah AWS tertentu, dan hal ini akan membantu mencegah sumber daya agar tidak dihapus, atau menonaktifkan tindakan layanan yang berpotensi berisiko. SCP yang Anda terapkan ke root organisasi Anda hanya akan memengaruhi akun-akun anggotanya saja, dan tidak akan berpengaruh pada akun manajemen. SCP hanya mengatur principal yang ada dalam organisasi Anda. SCP Anda tidak mengatur principal yang ada di luar organisasi Anda yang mengakses sumber daya Anda.

Jika Anda menggunakan AWS Control Tower, Anda dapat memanfaatkan kontrol dan zona landasannya sebagai dasar untuk pagar pembatas izin dan lingkungan multi-akun Anda. Zona landasan menyediakan lingkungan dasar yang telah dikonfigurasi sebelumnya dan aman dengan akun terpisah untuk beban kerja dan aplikasi yang berbeda-beda. Pagar pembatas memberlakukan kontrol wajib seputar keamanan, operasi, dan kepatuhan melalui kombinasi Kebijakan Kontrol Layanan (SCP), aturan AWS Config, dan konfigurasi lainnya. Namun, saat menggunakan pagar pembatas dan zona landasan Control Tower bersama SCP Organisasi kustom, sangat penting untuk mengikuti praktik terbaik yang diuraikan dalam dokumentasi AWS untuk menghindari konflik dan memastikan tata kelola yang tepat. Lihat panduan AWS Control Tower untuk AWS Organizations guna mengetahui rekomendasi terperinci tentang pengelolaan SCP, akun, dan unit organisasi (OU) dalam lingkungan Control Tower.

Dengan mematuhi pedoman ini, Anda dapat secara efektif memanfaatkan pagar pembatas, zona landasan, dan SCP kustom Control Tower sambil mengurangi potensi konflik serta memastikan tata kelola dan kontrol yang tepat atas lingkungan AWS multi-akun Anda.

Langkah selanjutnya adalah menggunakan kebijakan sumber daya IAM untuk mencakup tindakan-tindakan yang tersedia yang dapat diambil pada sumber daya yang mereka atur, bersama dengan

kondisi apa pun yang harus dipenuhi oleh principal. Ini bisa berupa memungkinkan semua tindakan selama principal adalah bagian dari organisasi Anda (menggunakan kunci kondisi PrincipalOrgId), atau sedetail hanya mengizinkan tindakan tertentu dengan peran IAM tertentu. Anda dapat mengambil pendekatan serupa terhadap kondisi dalam kebijakan kepercayaan peran IAM. Jika kebijakan kepercayaan sumber daya atau peran secara eksplisit menyebutkan suatu principal di akun yang sama sebagai peran atau sumber daya yang diaturnya, principal tersebut tidak perlu dilampiri dengan kebijakan IAM yang memberikan izin yang sama. Jika principal berada di akun yang berbeda dari sumber daya, maka principal memang perlu dilampiri dengan kebijakan IAM yang memberikan izin tersebut.

Sering kali, sebuah tim beban kerja ingin mengelola izin yang dibutuhkan beban oleh kerja mereka. Hal ini mungkin mengharuskan mereka membuat peran IAM dan kebijakan izin IAM yang baru. Anda dapat merekam cakupan maksimum izin yang boleh diberikan tim dalam batasan izin IAM, dan mengaitkan dokumen ini ke peran IAM yang kemudian dapat digunakan tim untuk mengelola peran IAM dan izin mereka. Pendekatan ini dapat memberi mereka fleksibilitas untuk menyelesaikan pekerjaan mereka sambil memitigasi risiko yang timbul karena memiliki akses administratif.

Langkah yang lebih terperinci adalah menerapkan teknik manajemen akses istimewa (PAM) dan manajemen akses tinggi sementara (TEAM). Salah satu contoh PAM adalah mewajibkan principal untuk melakukan autentikasi multi-faktor sebelum mengambil tindakan yang memerlukan hak akses istimewa. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA. TEAM memerlukan sebuah solusi yang mengelola persetujuan dan jangka waktu dari dan hingga kapan principal diizinkan memiliki akses yang ditingkatkan. Salah satu pendekatannya adalah menambahkan principal untuk sementara ke kebijakan kepercayaan peran untuk peran IAM yang memiliki akses yang ditingkatkan. Pendekatan lain adalah, dalam operasi normal, mengurangi izin yang diberikan kepada principal oleh peran IAM dengan menggunakan kebijakan sesi, dan kemudian mencabut sementara pembatasan ini selama jendela waktu yang disetujui. Untuk mempelajari lebih lanjut tentang solusi yang divalidasi oleh AWS dan mitra terpilih, lihat Akses yang ditingkatkan sementara.

Langkah-langkah implementasi

1. Isolasikan beban kerja dan lingkungan Anda ke dalam Akun AWS terpisah.
2. Gunakan SCP untuk mengurangi izin maksimum yang dapat diberikan kepada principal yang ada dalam akun anggota di organisasi Anda.
   a. Saat mendefinisikan SCP untuk mengurangi kumpulan izin maksimum yang dapat diberikan kepada pengguna utama dalam akun anggota organisasi Anda, Anda dapat memilih antara pendekatan daftar izinkan atau daftar tolak. Strategi daftar izinkan secara eksplisit menentukan

akses yang diizinkan dan secara implisit memblokir semua akses lainnya. Strategi daftar tolak secara eksplisit menentukan akses yang tidak diizinkan dan mengizinkan semua akses lainnya secara default. Kedua strategi ini memiliki kelebihan dan kompromi masing-masing, dan pilihan yang tepat akan tergantung pada persyaratan dan model risiko spesifik organisasi Anda. Untuk detail selengkapnya, lihat Strategi untuk menggunakan SCP.

   b. Selain itu, tinjau contoh kebijakan kontrol layanan untuk memahami cara menyusun SCP secara efektif.

3. Gunakan kebijakan sumber daya IAM untuk mengurangi cakupan dan menentukan kondisi untuk tindakan yang diizinkan pada sumber daya. Gunakan kondisi dalam kebijakan kepercayaan peran IAM untuk membuat batasan pada pengambilan peran.
4. Tetapkan batasan izin IAM ke peran IAM yang kemudian dapat digunakan tim beban kerja untuk mengelola peran IAM dan izin IAM beban kerja mereka sendiri.
5. Evaluasi solusi PAM dan TEAM berdasarkan kebutuhan Anda.

Sumber daya

Dokumen terkait:

- Perimeter data pada AWS
- Tetapkan pagar pembatas izin dengan menggunakan perimeter data
- Logika evaluasi kebijakan

Contoh terkait:

- Contoh kebijakan kontrol layanan

Alat terkait:

- Solusi AWS: Manajemen Akses Tinggi Sementara
- Solusi mitra keamanan tervalidasi untuk TEAM

SEC03-BP06 Mengelola akses berdasarkan siklus hidup

Lakukan pemantauan dan penyesuaian terhadap izin-izin yang diberikan kepada para principal Anda (pengguna, peran, dan grup) di sepanjang siklus hidupnya dalam organisasi Anda. Sesuaikan

keanggotaan grup jika pengguna berganti peran, dan hapus akses saat seorang pengguna meninggalkan organisasi.

Hasil yang diinginkan: Anda melakukan pemantauan dan menyesuaikan izin sepanjang siklus hidup principal yang ada dalam organisasi, mengurangi risiko hak akses istimewa yang tidak perlu. Anda memberikan akses yang sesuai saat membuat pengguna. Anda mengubah akses saat tanggung jawab pengguna berubah, dan Anda menghapus akses ketika pengguna tersebut tidak lagi aktif atau telah meninggalkan organisasi. Anda mengelola perubahan yang terjadi pada pengguna, peran, dan grup secara terpusat. Anda menggunakan otomatisasi untuk menyebarkan perubahan-perubahan ke lingkungan AWS Anda.

Anti-pola umum:

- Memberikan hak akses yang berlebihan atau luas ke identitas di awal, yang melebihi hak akses yang awalnya diperlukan.
- Tidak meninjau dan menyesuaikan hak akses saat peran dan tanggung jawab identitas berubah dari waktu ke waktu.
- Membiarkan identitas yang tidak aktif atau sudah dihentikan masih memiliki hak akses yang aktif. Hal ini akan meningkatkan risiko akses yang tidak sah.
- Tidak memanfaatkan otomatisasi untuk mengelola siklus hidup identitas.

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

Panduan implementasi

Kelola dan sesuaikan secara cermat hak akses yang Anda berikan kepada identitas (seperti pengguna, peran, grup) di sepanjang siklus hidup mereka. Siklus hidup ini mencakup fase onboarding awal, perubahan peran dan tanggung jawab yang berkelanjutan, dan akhirnya offboarding atau penghentian. Lakukan pengelolaan akses secara proaktif berdasarkan tahap siklus hidup untuk mempertahankan tingkat akses yang sesuai. Patuhi prinsip hak akses paling rendah untuk mengurangi munculnya risiko hak akses yang berlebihan atau tidak perlu.

Anda dapat mengelola siklus hidup pengguna IAM secara langsung dalam Akun AWS, ataupun melalui federasi dari penyedia identitas tenaga kerja Anda ke Pusat Identitas IAM AWS. Untuk pengguna IAM, Anda dapat membuat, memodifikasi, dan menghapus pengguna dan izin terkait mereka yang ada dalam Akun AWS. Untuk pengguna gabungan (federated user), Anda dapat menggunakan Pusat Identitas IAM untuk mengelola siklus hidup mereka dengan menyinkronkan

informasi pengguna dan grup dari penyedia identitas organisasi Anda dengan menggunakan protokol System for Cross-domain Identity Management (SCIM).

SCIM adalah sebuah protokol standar terbuka untuk penyediaan dan penghapusan penyediaan identitas pengguna otomatis di berbagai sistem. Dengan mengintegrasikan penyedia identitas Anda dengan Pusat Identitas IAM dengan menggunakan SCIM, Anda dapat secara otomatis melakukan sinkronisasi informasi pengguna dan grup, membantu memvalidasi bahwa hak akses diberikan, dimodifikasi, atau dicabut berdasarkan perubahan sumber identitas otoritatif yang ada di organisasi Anda.

Ketika peran dan tanggung jawab karyawan berubah dalam organisasi Anda, sesuaikan hak akses mereka sesuai keperluan. Anda dapat menggunakan kumpulan izin Pusat Identitas IAM untuk menentukan peran atau tanggung jawab pekerjaan yang berbeda-beda dan kemudian mengaitkannya dengan kebijakan IAM dan izin IAM yang sesuai. Saat peran seorang karyawan berubah, Anda dapat memperbarui kumpulan izin yang ditetapkan padanya untuk menyesuaikan dengan tanggung jawab baru mereka. Lakukan verifikasi bahwa mereka memiliki akses yang diperlukan dan sekaligus mematuhi prinsip hak akses paling rendah.

Langkah-langkah implementasi

1. Tentukan dan dokumentasikan proses siklus hidup manajemen akses, termasuk prosedur-prosedur yang harus dilakukan untuk memberikan akses awal, peninjauan berkala, dan offboarding.
2. Implementasikan peran IAM, grup, dan batasan izin untuk mengelola akses secara kolektif dan memberlakukan tingkat akses maksimum yang diizinkan.
3. Berintegrasi dengan sebuah penyedia identitas terfederasi (seperti Microsoft Active Directory, Okta, Ping Identity) sebagai sumber otoritatif untuk informasi pengguna dan grup menggunakan Pusat Identitas IAM.
4. Gunakan protokol SCIM untuk melakukan sinkronisasi informasi pengguna dan grup dari penyedia identitas ke Penyimpanan Identitas Pusat Identitas IAM.
5. Buat kumpulan izin di Pusat Identitas IAM yang merepresentasikan peran atau tanggung jawab pekerjaan yang berbeda-beda dalam organisasi Anda. Tentukan kebijakan IAM dan izin IAM yang sesuai untuk masing-masing kumpulan izin.
6. Implementasikan peninjauan akses secara rutin, pencabutan akses yang cepat, dan peningkatan berkelanjutan terhadap proses siklus hidup manajemen akses.
7. Berikan pelatihan dan pengetahuan kepada karyawan tentang praktik terbaik manajemen akses.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC02-BP04 Mengandalkan penyedia identitas tersentralisasi

Dokumen terkait:

- Kelola sumber identitas Anda
- Kelola identitas di Pusat Identitas IAM
- Menggunakan AWS Identity and Access Management Access Analyzer
- Pembuatan kebijakan IAM Access Analyzer

Video terkait:

- AWS re:Inforce 2023 - Kelola akses tambahan sementara dengan Pusat Identitas Pusat Identitas IAM AWS
- AWS re:invent 2022 - Menyederhanakan akses tenaga kerja Anda dengan Pusat Identitas IAM
- AWS re:invent 2022 - Memanfaatkan kekuatan kebijakan IAM & mengendalikan izin dengan Access Analyzer

SEC03-BP07 Menganalisis akses publik dan lintas akun

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 sumber daya AWS Anda yang mana yang dibagikan dan dengan siapa. Lakukan pemantauan dan audit secara terus-menerus terhadap sumber daya bersama untuk memastikan bahwa sumber daya tersebut hanya dibagikan kepada principal yang sah.

Anti-pola umum:

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

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

Panduan implementasi

Jika akun Anda berada di AWS Organizations, maka Anda dapat memberikan akses sumber daya ke seluruh organisasi, unit organisasi tertentu, atau masing-masing akun individu. Jika akun Anda bukan merupakan anggota suatu organisasi, maka Anda dapat berbagi sumber daya dengan akun individu. Anda dapat memberikan akses lintas akun secara langsung dengan menggunakan kebijakan berbasis sumber daya — misalnya, kebijakan bucket Amazon Simple Storage Service (Amazon S3) — atau dengan mengizinkan principal di akun lain untuk mengambil peran IAM di akun Anda. Saat menggunakan kebijakan sumber daya, verifikasi bahwa akses tersebut hanya diberikan kepada principal yang sah. Tentukan sebuah 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 akun tersebut. Keamanan tersebut akan meninjau kebijakan sumber daya secara terus-menerus, dan melaporkan temuan akses lintas akun dan publik untuk memudahkan Anda dalam menganalisis potensi akses yang meluas. Pertimbangkan untuk mengonfigurasi IAM Access Analyzer dengan AWS Organizations untuk memastikan Anda memiliki visibilitas tentang semua akun Anda. IAM Access Analyzer juga memungkinkan Anda untuk melihat pratinjau temuan sebelum melakukan deployment atas izin sumber daya. Hal ini akan memungkinkan Anda untuk memvalidasi bahwa perubahan kebijakan hanya memberikan akses lintas akun dan publik yang benar-benar diinginkan ke sumber daya Anda. Saat merancang untuk akses multi-akun, Anda dapat menggunakan kebijakan kepercayaan untuk mengontrol dalam kasus apa sebuah peran bisa diambil. Misalnya, Anda dapat menggunakan kunci kondisi `PrincipalOrgId` untuk menolak upaya pengambilan peran dari luar AWS Organizations Anda.

AWS Config dapat melaporkan sumber daya yang salah konfigurasi, dan melalui pemeriksaan kebijakan AWS Config, dapat mendeteksi sumber daya yang memiliki akses publik yang dikonfigurasi. Layanan-layanan seperti AWS Control Tower dan AWS Security Hub menyederhanakan deployment pemeriksaan dan pagar pembatas di seluruh AWS Organizations untuk mengidentifikasi dan memperbaiki sumber daya yang terpapar publik. Misalnya, AWS Control Tower memiliki sebuah pagar pembatas terkelola yang dapat mendeteksi jika ada snapshot Amazon EBS yang dapat dipulihkan oleh Akun AWS.

Langkah-langkah implementasi

- Pertimbangkan untuk menggunakanAWS Config for AWS Organizations: AWS Config akan memungkinkan Anda mengumpulkan temuan dari beberapa akun dalam sebuah AWS

Organizations ke akun administrator yang didelegasikan. Ini akan memberikan tampilan yang komprehensif, dan memungkinkan Anda untuk melakukan deployment terhadap 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 akan membantu Anda mengidentifikasi sumber daya di akun dan organisasi Anda, seperti bucket Amazon S3 atau peran IAM yang dibagikan ke entitas eksternal.
- Gunakan remediasi otomatis AWS Config untuk merespons perubahan konfigurasi akses publik bucket Amazon S3: Anda dapat secara otomatis mengaktifkan pengaturan blokir akses publik untuk bucket Amazon S3.
- Terapkan pemantauan dan peringatan untuk mengidentifikasi apakah bucket Amazon S3 telah menjadi publik: Anda harus memiliki pemantauan dan peringatan yang tersedia untuk mengidentifikasi kapan Blokir Akses Publik Amazon S3 dimatikan, dan apakah bucket Amazon S3 sudah publik atau tidak. Selain itu, jika Anda menggunakan AWS Organizations, Anda dapat membuat sebuah kebijakan kontrol layanan yang mencegah perubahan pada kebijakan akses publik Amazon S3. AWS Trusted Advisor akan memeriksa 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 yang dalam sebuah bucket. Pemeriksaan Trusted Advisor memeriksa izin bucket eksplisit dan kebijakan-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, lihat Cara Menggunakan AWS Config untuk Memantau dan Merespons Bucket Amazon S3 yang Memungkinkan Akses Publik.

Saat meninjau kontrol akses untuk bucket Amazon S3, penting untuk mempertimbangkan sifat data yang tersimpan di dalamnya. Amazon Macie adalah layanan yang dirancang untuk membantu Anda menemukan dan melindungi data sensitif, seperti Informasi Pengenal Pribadi (PII), Informasi Kesehatan yang Dilindungi (PHI), dan kredensial seperti kunci pribadi atau kunci akses AWS.

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 di Semua Akun di Organisasi Anda
- AWS Config dan AWS Organizations
- Membuat AMI Anda tersedia secara umum untuk digunakan di Amazon EC2

Video terkait:

- Praktik Terbaik untuk mengamankan lingkungan multiakun Anda
- Memahami lebih dalam IAM Access Analyzer

SEC03-BP08 Membagikan sumber daya secara aman dalam organisasi Anda

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

Hasil yang diinginkan: Mengurangi akses yang tidak diinginkan sekecil hingga sekecil mungkin dengan menggunakan metode yang aman untuk berbagi sumber daya dalam organisasi Anda, dan membantu inisiatif pencegahan kehilangan data Anda. Mengurangi overhead operasional daripada mengelola komponen satu per satu, akan mengurangi kesalahan yang diakibatkan oleh 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 sebuah komponen tidak diperlukan lagi. Untuk panduan preskriptif tentang cara melakukan analisis sumber daya bersama secara eksternal, silakan lihat SEC03-BP07 Menganalisis akses publik dan lintas akun.

Anti-pola umum:

- Tidak ada proses untuk melakukan pemantauan secara terus-menerus dan dan memberikan peringatan otomatis mengenai 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 yang tumpang tindih secara manual, saat diperlukan.

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

Panduan implementasi

Rancang arsitektur pola dan kontrol akses Anda untuk mengelola penggunaan sumber daya yang dibagikan secara aman dan hanya dengan entitas-entitas yang tepercaya. Lakukan pemantauan terhadap sumber daya yang dibagikan dan peninjauan terhadap akses sumber daya yang dibagikan secara terus-menerus serta tetap waspada terhadap adanya pembagian yang tidak terduga atau tidak tepat. Tinjau Analisis akses publik dan akses lintas akun untuk membantu Anda menetapkan tata kelola untuk mengurangi akses eksternal hanya ke sumber daya yang memerlukannya, dan untuk membuat proses untuk melakukan pemantauan secara terus menerus dan memberi peringatan secara otomatis.

Berbagi lintas akun dalam AWS Organizations didukung oleh sejumlah layanan AWS, seperti AWS Security Hub, Amazon GuardDuty, dan AWS Backup. Layanan-layanan ini akan 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 sebuah 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 berbagi sumber daya umum lainnya, seperti subnet VPC dan lampiran Gateway Transit, AWS Network Firewall, atau Amazon SageMaker AI pipelines.

Untuk membatasi akun Anda agar hanya berbagi sumber daya dalam organisasi Anda, gunakan kebijakan kontrol layanan (SCP) untuk mencegah akses ke principal eksternal. Saat berbagi sumber daya, gabungkan kontrol berbasis identitas dan kontrol jaringan untuk membuat perimeter data bagi organisasi Anda guna membantu Anda melindungi dari akses yang tidak diinginkan. Perimeter data adalah kumpulan pagar pembatas preventif untuk membantu Anda memverifikasi bahwa hanya identitas yang Anda percaya saja yang mengakses sumber daya tepercaya dari jaringan yang dikenal. Kontrol ini akan menetapkan batas yang sesuai terkait sumber daya apa yang dapat dibagikan, serta mencegah dibagikannya atau bocornya sumber daya yang tidak semestinya terjadi.

Misalnya, sebagai bagian dari perimeter data, Anda dapat menggunakan kebijakan titik akhir VPC dan kondisi `AWS:PrincipalOrgId` untuk memastikan identitas yang mengakses bucket Amazon S3 yang dimiliki organisasi Anda. Penting untuk dicatat bahwa SCP tidak berlaku untuk peran terkait layanan atau principal layanan AWS.

Saat menggunakan Amazon S3, matikan ACL untuk bucket Amazon S3 Anda dan gunakan kebijakan IAM untuk menentukan kontrol akses. Untuk membatasi akses ke asal Amazon S3 dari Amazon CloudFront, bermigrasilah dari identitas akses asal (OAI) ke kontrol akses asal (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 cara mengelola izin untuk berbagi sumber daya secara eksternal, lihat Manajemen izin.

Langkah-langkah implementasi

1. Gunakan AWS Organizations: AWS Organizations adalah layanan manajemen akun yang akan memungkinkan Anda mengonsolidasikan 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-kebijakan yang berbeda ke setiap OU untuk membantu Anda 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-layanan AWS yang terintegrasi dengan Organisasi.
2. Integrasikan AWS Organizations dengan layanan AWS: Ketika Anda menggunakan sebuah layanan AWS untuk melakukan tugas atas nama Anda di akun anggota organisasi Anda, AWS Organizations akan membuat peran terkait layanan (SLR) 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 cara mengaktifkan akses tepercaya, lihat Menggunakan AWS Organizations dengan layanan AWS dan layanan AWS lain yang dapat Anda gunakan dengan Organisasi.
3. Tetapkan perimeter data: Perimeter data memberikan batas kepercayaan dan kepemilikan yang jelas. Di AWS, hal ini biasanya direpresentasikan sebagai organisasi AWS Anda yang dikelola oleh AWS Organizations, bersama dengan jaringan atau sistem on-premise yang mengakses sumber daya AWS Anda. Perimeter data bertujuan untuk memverifikasi bahwa akses diizinkan jika identitasnya dipercaya, sumber dayanya dipercaya, dan jaringannya dikenal. Namun, menetapkan perimeter data bukanlah pendekatan yang bisa digunakan untuk semua situasi. Evaluasi dan adopsi tujuan kontrol yang diuraikan dalam laporan resmi Membangun

Perimeter di AWS berdasarkan model dan persyaratan risiko keamanan spesifik Anda. Anda harus mempertimbangkan postur risiko yang Anda miliki dengan cermat dan menerapkan kontrol perimeter yang selaras dengan kebutuhan keamanan Anda.

4. Gunakan berbagi sumber daya di layanan AWS dan terapkan pembatasan dengan sesuai: Banyak layanan AWS memungkinkan Anda berbagi sumber daya dengan akun lain, atau menargetkan sumber daya di akun lain, seperti Amazon Machine Image (AMI) dan AWS Resource Access Manager (AWS RAM). Batasi API `ModifyImageAttribute` untuk menentukan akun tepercaya yang akan berbagi AMI. Tentukan kondisi `ram:RequestedAllowsExternalPrincipals` saat menggunakan AWS RAM untuk membatasi berbagi ke organisasi Anda saja, untuk membantu mencegah akses dari identitas yang tidak tepercaya. Untuk panduan dan pertimbangan preskriptif, lihat Berbagi sumber daya dan target eksternal.
5. Gunakan AWS RAM untuk berbagi secara aman dengan sebuah akun atau dengan Akun AWS lainnya: AWS RAM akan membantu Anda secara aman membagikan sumber daya yang Anda buat kepada peran dan pengguna di akun Anda, serta dengan Akun AWS lainnya. Dalam lingkungan multiakun, AWS RAM akan memungkinkan Anda untuk membuat sumber daya satu kali dan membagikannya ke 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 Anda memiliki sumber daya yang Anda bagikan sebelumnya dengan menggunakan kebijakan berbasis sumber daya, maka Anda dapat menggunakan API `PromoteResourceShareCreatedFromPolicy` atau yang setara untuk mempromosikan pembagian sumber daya ke pembagian sumber daya AWS RAM penuh.

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

Sumber daya

Praktik-praktik terbaik terkait:

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

Dokumen terkait:

- Pemilik bucket yang memberikan izin lintas akun untuk objek yang bukan miliknya
- Cara menggunakan Kebijakan Kepercayaan dengan IAM
- Membangun Perimeter Data di AWS
- Cara menggunakan ID eksternal saat memberikan pihak ketiga akses ke sumber daya AWS Anda
- Layanan AWS yang dapat Anda gunakan dengan AWS Organizations
- Membuat perimeter data pada AWS: Izinkan hanya identitas tepercaya saja yang bisa 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 dengan menggunakan prinsip hak akses paling rendah dengan kredensial sementara. Melalui kerja sama dengan pihak ketiga, Anda dapat mengurangi cakupan dampak dan risiko yang mungkin dimunculkan oleh akses yang tidak diinginkan.

Hasil yang diinginkan: Anda menghindari penggunaan kredensial AWS Identity and Access Management (IAM) jangka panjang seperti kunci akses dan kunci rahasia karena akan menimbulkan risiko keamanan jika disalahgunakan. Sebagai gantinya, Anda menggunakan peran IAM dan kredensial sementara untuk meningkatkan postur keamanan Anda dan meminimalkan overhead operasional untuk mengelola kredensial jangka panjang. Saat memberikan akses kepada pihak ketiga, Anda menggunakan pengidentifikasi unik universal (UUID) sebagai ID eksternal dalam kebijakan kepercayaan IAM dan menjaga kebijakan IAM terlampir pada peran di bawah kendali Anda

untuk memastikan hak akses paling rendah. Untuk panduan preskriptif tentang menganalisis sumber daya yang dibagikan secara eksternal, lihat SEC03-BP07 Menganalisis akses publik dan akses lintas akun.

Anti-pola 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 diterapkan: 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 sebuah solusi pemantauan yang perlu mengakses sumber daya yang ada 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 dengan menggunakan kondisi ID eksternal. Saat menggunakan ID eksternal, Anda atau pihak ketiga dapat membuat sebuah ID unik untuk setiap pelanggan, pihak ketiga, atau penghunian. Setelah dibuat, ID unik tersebut tidak boleh dikontrol oleh siapa pun selain Anda. Pihak ketiga harus mengimplementasikan sebuah proses untuk memberikan ID eksternal melalui cara yang aman, dapat diaudit, dan diproduksi kembali.

Anda juga dapat menggunakan IAM Roles Anywhere untuk 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. Ketahui selalu layanan AWS lain yang mendukung fitur berbagi, seperti AWS Well-Architected Tool yang memungkinkan berbagi beban kerja dengan Akun AWS lain, dan AWS Resource Access Manager yang membantu Anda secara aman membagikan sumber daya AWS yang Anda miliki ke akun lain.

Langkah-langkah implementasi

1. Gunakan peran lintas akun untuk memberikan akses ke akun eksternal. Peran lintas akun akan mengurangi jumlah informasi sensitif yang disimpan oleh akun eksternal dan pihak ketiga yang diperlukan untuk melayani pelanggan mereka. Peran lintas akun akan memungkinkan Anda

memberikan akses ke sumber daya AWS yang ada di akun Anda kepada pihak ketiga secara aman, seperti Partner AWS atau akun-akun lainnya yang ada di organisasi Anda, dan Anda pun tetap dapat mengelola dan mengaudit akses tersebut. Pihak ketiga mungkin memberikan layanan kepada Anda dari sebuah infrastruktur hibrida atau menarik data ke lokasi di luar situs. IAM Roles Anywhere akan membantu Anda mengizinkan beban kerja pihak ketiga berinteraksi secara aman dengan beban kerja AWS Anda dan makin mengurangi kebutuhan akan kredensial jangka panjang.

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

2. Lakukan uji tuntas dan pastikan akses aman untuk penyedia SaaS pihak ketiga. Saat berbagi sumber daya dengan penyedia SaaS pihak ketiga, lakukan uji tuntas menyeluruh guna memastikan mereka memiliki pendekatan yang aman dan bertanggung jawab untuk mengakses sumber daya AWS Anda. Evaluasi model tanggung jawab bersama yang mereka miliki untuk memahami langkah-langkah keamanan yang mereka terapkan dan aspek keamanan yang menjadi tanggung jawab Anda. Pastikan bahwa penyedia SaaS memiliki proses yang aman dan dapat diaudit untuk mengakses sumber daya Anda, termasuk penggunaan ID eksternal dan prinsip akses hak akses paling rendah. Penggunaan ID eksternal membantu mengatasi masalah "confused deputy".

   Implementasikan kontrol keamanan untuk memastikan akses yang aman dan kepatuhan terhadap prinsip hak akses paling rendah saat memberikan akses kepada penyedia SaaS pihak ketiga. Hal ini mungkin mencakup penggunaan ID eksternal, pengidentifikasi unik universal (UUID), dan kebijakan kepercayaan IAM yang membatasi akses hanya ke hal yang benar-benar diperlukan. Bekerjasamalah secara erat dengan penyedia SaaS untuk membangun mekanisme akses yang aman, tinjau akses mereka ke sumber daya AWS Anda secara teratur, dan lakukan audit untuk memastikan kepatuhan dengan persyaratan keamanan Anda.
3. Menghilangkan 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, buatlah sebuah rencana untuk bermigrasi ke akses berbasis peran. Untuk mendapatkan detail tentang cara mengelola kunci, lihat Manajemen identitas. Selain itu, bekerjasamalah dengan tim Akun AWS Anda dan pihak ketiga untuk menyusun runbook mitigasi risiko. Untuk panduan preskriptif mengenai cara merespons dan memitigasi potensi dampak insiden keamanan, lihat Respons insiden.
4. Verifikasi bahwa pengaturan memiliki panduan preskriptif atau otomatis. ID eksternal bukan sesuatu yang rahasia, tetapi ID eksternal tidak boleh berupa nilai yang mudah ditebak, seperti

nomor telepon, nama, atau ID akun. Buatlah ID eksternal menjadi bidang hanya baca sehingga ID eksternal tersebut tidak dapat diubah untuk tujuan meniru penyiapan.

Anda atau pihak ketiga dapat membuat ID eksternal. Bentuklah sebuah 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.

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

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

5. Akun untuk perubahan. Struktur akun Anda, kebutuhan Anda terhadap 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. Lakukan audit tingkat akses yang Anda berikan secara berkala, dan terapkan metode deteksi yang akan memberikan Anda peringatan 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 terjadi 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 dampaknya terhadap sumber daya lainnya.

   Untuk panduan preskriptif mengenai metode deteksi, silakan lihat Praktik terbaik deteksi.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC02-BP02 Menggunakan kredensial sementara

- SEC03-BP05 Tentukan pagar pembatas izin untuk organisasi Anda
- SEC03-BP06 Mengelola akses berdasarkan siklus hidup
- SEC03-BP07 Menganalisis akses publik dan lintas akun
- SEC04 Deteksi

Dokumen terkait:

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

Video terkait:

- Bagaimana cara mengizinkan pengguna atau peran yang ada dalam akses Akun AWS terpisah ke Akun AWS saya?
- AWS re:Invent 2018: Menjadi Master Kebijakan IAM dalam 60 Menit atau Kurang
- Pusat Pengetahuan AWS Langsung: Praktik Terbaik dan Keputusan Desain IAM

Contoh terkait:

- Mengonfigurasikan akses lintas akun ke Amazon DynamoDB
- Alat Kueri Jaringan AWS STS

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 untuk memungkinkan 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
- SEC04-BP02 Catat log, temuan, dan metrik di lokasi standar
- SEC04-BP03 Korelasikan dan perkaya data peringatan keamanan
- SEC04-BP04 Mulai melakukan remediasi untuk sumber daya yang tidak mematuhi persyaratan

SEC04-BP01 Mengonfigurasi pencatatan log layanan dan aplikasi

Pertahankan log peristiwa keamanan dari layanan dan aplikasi. 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: Organisasi harus dapat secara andal dan konsisten mengambil log peristiwa keamanan dari layanan dan aplikasi AWS secara tepat waktu ketika diperlukan untuk memenuhi proses atau kewajiban internal, misalnya untuk respons insiden keamanan. Sebaiknya pusatkan log untuk mendapatkan hasil operasional yang lebih baik.

Anti-pola 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 menerapkan praktik terbaik ini: Menerapkan mekanisme analisis akar penyebab (RCA) untuk insiden keamanan dan sumber bukti untuk kewajiban tata kelola, risiko, dan kepatuhan Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: 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 lini masa insiden. Log juga diperlukan untuk pembuatan peringatan yang mengindikasikan bahwa

tindakan-tindakan tertentu telah terjadi. Sangat penting bagi Anda untuk memilih, menyalakan, menyimpan, dan menyiapkan mekanisme kueri dan pengambilan serta pembuatan peringatan.

Langkah-langkah implementasi

- Pilih dan gunakan sumber log. Sebelum melakukan penyelidikan keamanan, Anda perlu mengambil log yang relevan untuk merekonstruksi aktivitas secara surut di Akun AWS. Pilih 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 dengan menggunakan AWS CloudTrail atau jejak AWS Organizations, dan konfigurasikan bucket Amazon S3 untuk jejak tersebut.

  AWS CloudTrail adalah sebuah layanan pencatatan log yang melacak panggilan API yang dibuat terhadap Akun AWS yang merekam aktivitas layanan AWS. Hal ini dinyalakan secara default dengan retensi 90 hari dari peristiwa manajemen yang dapat diambil melalui fasilitas Riwayat Peristiwa CloudTrail menggunakan AWS Management Console, AWS CLI, atau SDK AWS. Untuk retensi dan visibilitas peristiwa data yang lebih lama, Anda perlu membuat CloudTrail Trail dan kaitkan dengan bucket Amazon S3, dan juga dengan grup log Amazon CloudWatch. Selain itu, Anda dapat membuat CloudTrail Lake, yang menyimpan dan mempertahankan log CloudTrail hingga tujuh tahun dan menyediakan fasilitas kueri berbasis SQL

  AWS merekomendasikan agar pelanggan yang menggunakan lalu lintas jaringan berkemampuan VPC dan log DNS yang menggunakan Log Alur VPC dan log kueri Amazon Route 53 Resolver untuk mengalirkannya ke sebuah 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, menormalkan, dan menyimpan data log ini dalam format Apache Parquet dan Open Cybersecurity Schema Framework (OCSF), yang siap untuk pelaksanaan kueri. Danau Keamanan juga mendukung log AWS lainnya dan log dari sumber pihak ketiga.

  Layanan-layanan AWS dapat menghasilkan log yang tidak ditangkap oleh sumber log dasar, seperti log Penyeimbangan Beban Elastis, 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 melihat daftar lengkap opsi pencatatan dan

pemantauan, silakan lihat Lampiran A: Definisi kemampuan cloud — Pencatatan Log dan Peristiwa yang ada di Panduan Respons Insiden Keamanan AWS.

- Kemampuan pencatatan log penelitian untuk masing-masing layanan dan aplikasi AWS: Setiap layanan dan aplikasi AWS memberi Anda opsi untuk penyimpanan log, yang masing-masing memiliki kemampuan retensi dan siklus hidupnya sendiri. Dua layanan penyimpanan log yang paling umum adalah layanan Amazon Simple Storage Service (Amazon S3) dan Amazon CloudWatch. Untuk periode retensi yang panjang, sebaiknya gunakan Amazon S3 untuk mendapatkan 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 terkait dengan alat kueri yang Anda gunakan, kemampuan retensi, pemahaman, 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 dalam bucket Amazon S3 dapat dikueri secara native dengan menggunakan layanan-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 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, para 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 Anda serta gabungan mandat hukum, peraturan, dan bisnis.
- Gunakan pencatatan log untuk masing-masing layanan dan aplikasi AWS yang memiliki kebijakan retensi dan siklus hidup yang tepat: Untuk setiap layanan atau aplikasi AWS yang ada di organisasi Anda, cari panduan konfigurasi pencatatan log yang spesifik untuk aplikasi atau layanan tersebut:
  - Konfigurasikan Jejak AWS CloudTrail
  - Konfigurasikan Amazon VPC Flow Logs
  - Konfigurasikan Ekspor Temuan Amazon GuardDuty
  - Konfigurasikan perekaman AWS Config
  - Konfigurasikan lalu lintas ACL web AWS WAF
  - Konfigurasikan log lalu lintas jaringan AWS Network Firewall

  - Konfigurasikan log akses Penyeimbangan Beban Elastis
  - Konfigurasikan log kueri Amazon Route 53 Resolver
  - Konfigurasikan log Amazon RDS
  - Konfigurasikan log Bidang Kontrol Amazon EKS
  - Konfigurasikan agen Amazon CloudWatch untuk instans Amazon EC2 dan server on-premise
- Pilih dan terapkan mekanisme kueri untuk log: Untuk kueri log, Anda dapat menggunakan CloudWatch Logs Insights untuk data yang disimpan di grup log CloudWatch, serta Amazon Athena dan Amazon OpenSearch Service untuk data yang disimpan di Amazon S3. Anda dapat menggunakan alat kueri pihak ketiga, misalnya sebuah layanan informasi keamanan dan manajemen peristiwa (SIEM).

  Proses untuk memilih alat kueri log harus mempertimbangkan aspek orang, proses, dan teknologi dalam operasi keamanan Anda. Pilihlah sebuah alat yang memenuhi persyaratan operasional, bisnis, dan keamanan, serta dapat diakses dan dipelihara dalam jangka panjang. Perlu diingat bahwa alat kueri log bekerja secara optimal ketika jumlah log yang akan dipindai tidak melebihi batas alat. Tidak jarang terdapat beberapa alat kueri karena adanya kendala biaya atau teknis.

  Misalnya, Anda mungkin menggunakan alat manajemen informasi dan peristiwa keamanan (SIEM) 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 akan meminimalkan jumlah alat yang diperlukan untuk memaksimalkan efisiensi operasional, khususnya selama penyelidikan peristiwa keamanan.
- Gunakan log untuk pembuatan peringatan: AWS menyediakan peringatan melalui beberapa layanan keamanan:
  - AWS Config memantau dan merekam konfigurasi sumber daya AWS Anda serta memungkinkan Anda untuk mengotomatisasi evaluasi dan perbaikan berdasarkan konfigurasi yang diinginkan.
  - Amazon GuardDuty adalah sebuah layanan deteksi ancaman yang terus memantau aktivitas berbahaya dan perilaku tidak terotorisasi untuk melindungi Akun AWS dan beban kerja Anda. GuardDuty menyerap, mengumpulkan, dan menganalisis informasi dari sumber, seperti peristiwa manajemen dan data AWS CloudTrail, log DNS, Log Aliran VPC, dan log Audit Amazon EKS. GuardDuty menarik aliran data independen secara langsung dari CloudTrail, Log Aliran 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 dan mempertahankan log tersebut untuk tujuan penyelidikan dan kepatuhan.

- AWS Security Hub menyediakan satu tempat yang mengumpulkan, mengatur, dan memprioritaskan peringatan keamanan Anda, atau temuan, dari beberapa layanan AWS serta produk pihak ketiga opsional untuk memberikan Anda tampilan peringatan keamanan dan status kepatuhan secara komprehensif.

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

Sumber daya

Praktik-praktik terbaik terkait:

- SEC04-BP02 Catat log, temuan, dan metrik di lokasi standar
- SEC07-BP04 Tentukan manajemen siklus hidup data yang dapat diskalakan
- SEC10-BP06 Melakukan deployment alat di awal

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS
- Memulai dengan Danau Keamanan Amazon
- Memulai Log Amazon CloudWatch

Video terkait:

- AWS re:Invent 2022 - Memperkenalkan Danau Keamanan Amazon

Contoh terkait:

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

SEC04-BP02 Catat log, temuan, dan metrik di lokasi standar

Tim keamanan mengandalkan log dan temuan untuk melakukan analisis terhadap peristiwa yang mungkin mengindikasikan aktivitas yang tidak sah atau perubahan yang tidak disengaja. Untuk menyederhanakan analisis ini, lakukan perekaman log dan temuan keamanan di lokasi yang terstandardisasi. Hal ini membuat titik data yang menjadi perhatian tersedia untuk korelasi dan dapat menyederhanakan integrasi alat.

Hasil yang diinginkan: Anda memiliki pendekatan standar untuk mengumpulkan, menganalisis, dan memvisualisasikan data log, temuan, dan metrik. Tim keamanan dapat secara efisien mengorelasikan, menganalisis, dan memvisualisasikan data keamanan di seluruh sistem yang berbeda untuk menemukan kemungkinan adanya potensi peristiwa keamanan dan mengidentifikasi anomali. Sistem manajemen informasi dan peristiwa keamanan (SIEM) atau mekanisme lainnya diintegrasikan untuk melakukan kueri dan analisis data log guna melakukan respons, pelacakan, dan eskalasi peristiwa keamanan secara tepat waktu.

Anti-pola umum:

- Tim secara mandiri memiliki dan mengelola pencatatan log dan pengumpulan metrik yang tidak konsisten dengan strategi pencatatan log organisasi.
- Tim tidak memiliki kontrol akses yang memadai untuk membatasi visibilitas dan perubahan terhadap data yang dikumpulkan.
- Tim tidak mengatur log, temuan, dan metrik keamanan mereka sebagai bagian dari kebijakan klasifikasi data.
- Tim mengabaikan persyaratan kedaulatan dan pelokalan data saat melakukan konfigurasi terhadap pengumpulan data.

Manfaat menjalankan praktik terbaik ini: Solusi pencatatan log standar untuk mengumpulkan dan menanyakan data dan peristiwa log akan meningkatkan wawasan yang diperoleh dari informasi yang dikandungnya. Konfigurasi siklus hidup otomatis untuk data log yang dikumpulkan dapat mengurangi biaya yang ditimbulkan oleh penyimpanan log. Anda dapat membuat kontrol akses terperinci untuk informasi log yang dikumpulkan sesuai dengan sensitivitas data dan pola akses yang dibutuhkan oleh tim-tim Anda. Anda dapat mengintegrasikan peralatan untuk mengorelasikan, memvisualisasikan, dan memperoleh wawasan dari data.

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

Panduan implementasi

Pertumbuhan penggunaan AWS dalam sebuah organisasi menghasilkan peningkatan jumlah beban kerja dan lingkungan yang terdistribusi. Karena masing-masing beban kerja dan lingkungan ini menghasilkan data tentang aktivitas di dalamnya, pencatatan dan penyimpanan data ini secara lokal akan menimbulkan kesulitan untuk operasi keamanan. Tim keamanan menggunakan alat-alat seperti, misalnya sistem manajemen informasi dan peristiwa keamanan (SIEM), untuk mengumpulkan data dari sumber terdistribusi serta menjalani alur kerja korelasi, analisis, dan respons. Hal ini membutuhkan pengelolaan serangkaian izin yang kompleks untuk mengakses berbagai sumber data dan overhead tambahan dalam mengoperasikan proses-proses extract, transform, and load (ETL).

Untuk mengatasi tantangan ini, pertimbangkan menggabungkan semua sumber data log keamanan yang relevan ke dalam akun Arsip Log seperti yang dijelaskan dalam Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun. Hal ini mencakup semua data terkait keamanan dari beban kerja dan log yang dihasilkan layanan AWS, seperti, AWS CloudTrail, AWS WAF, Penyeimbangan Beban Elastis, dan Amazon Route 53. Ada beberapa manfaat untuk merekam data ini di lokasi terstandardisasi dalam sebuah Akun AWS terpisah dengan izin lintas akun yang tepat. Praktik ini membantu mencegah log dimanipulasi dalam beban kerja dan lingkungan yang mengalami kebocoran keamanan, menyediakan satu titik integrasi untuk alat tambahan, dan menawarkan model yang lebih sederhana untuk mengonfigurasi retensi data dan siklus hidup data. Evaluasi dampak kedaulatan data, cakupan kepatuhan, dan peraturan lainnya untuk menentukan apakah beberapa lokasi penyimpanan data dan periode retensi data keamanan diperlukan.

Untuk memudahkan perekaman dan standardisasi log dan temuan, lakukan evaluasi terhadap Amazon Security Lake di akun Arsip Log Anda. Anda dapat mengonfigurasi Danau Keamanan untuk secara otomatis menyerap data dari sumber umum seperti CloudTrail, Route 53, Amazon EKS, dan VPC Flow Logs. Anda juga dapat mengonfigurasi AWS Security Hub sebagai sumber data ke Danau Keamanan, memungkinkan Anda untuk mengkorelasikan temuan dari layanan-layanan AWS lain, seperti Amazon GuardDuty dan Amazon Inspector, dengan data log Anda. Anda juga dapat menggunakan integrasi sumber data pihak ketiga, atau mengonfigurasi sumber data kustom. Semua integrasi menstandardisasi data Anda ke dalam format Open Cybersecurity Schema Framework (OCSF), dan disimpan dalam bucket Amazon S3 sebagai file Parket, sehingga tidak perlu ada pemrosesan ETL.

Menyimpan data keamanan di lokasi terstandardisasi akan memberikan kemampuan analitik tingkat lanjut. AWS merekomendasikan Anda untuk men-deploy alat untuk analitik keamanan yang beroperasi di sebuah lingkungan AWS ke akun Peralatan Keamanan yang terpisah dari akun

Arsip Log Anda. Pendekatan ini akan memungkinkan Anda untuk mengimplementasikan kontrol secara mendalam guna melindungi integritas dan ketersediaan log serta proses manajemen log, terpisah dari alat yang mengaksesnya. Pertimbangkan untuk menggunakan layanan-layanan, seperti Amazon Athena, untuk menjalankan kueri sesuai permintaan yang menghubungkan beberapa sumber data. Anda juga dapat mengintegrasikan alat-alat visualisasi, seperti QuickSight. Solusi yang didukung AI makin banyak tersedia dan dapat melakukan berbagai fungsi, misalnya menerjemahkan temuan ke dalam ringkasan yang dapat dibaca manusia dan interaksi bahasa yang alami. Solusi ini sering kali lebih mudah diintegrasikan dengan memiliki lokasi penyimpanan data terstandardisasi untuk melakukan kueri.

Langkah-langkah implementasi

1. Buat akun Arsip Log dan Peralatan Keamanan
   a. Dengan menggunakan AWS Organizations, buat akun Arsip Log dan Peralatan Keamanan di bawah sebuah unit organisasi keamanan. Jika Anda menggunakan AWS Control Tower untuk mengelola organisasi Anda, maka akun Arsip Log dan Alat Keamanan akan dibuat untuk Anda secara otomatis. Konfigurasikan peran dan izin untuk mengakses dan mengelola akun ini sesuai kebutuhan.
2. Konfigurasikan lokasi data keamanan terstandardisasi Anda
   a. Tentukan strategi Anda untuk membuat lokasi data keamanan terstandardisasi. Anda dapat mencapai ini melalui opsi seperti pendekatan arsitektur danau data umum, produk data pihak ketiga, atau Amazon Security Lake. AWS merekomendasikan agar Anda merekam data keamanan dari Wilayah AWS yang ikut serta untuk akun Anda, bahkan ketika tidak digunakan secara aktif.
3. Konfigurasikan publikasi sumber data ke lokasi terstandardisasi Anda
   a. Identifikasi sumber-sumber untuk data keamanan Anda dan konfigurasikan untuk dipublikasikan ke lokasi terstandardisasi Anda. Lakukan evaluasi terhadap opsi-opsi untuk mengekspor data secara otomatis dalam format yang diinginkan dan bukan opsi-opsi di mana proses ETL perlu dikembangkan. Dengan Amazon Security Lake, Anda dapat mengumpulkan data dari sumber AWS yang didukung dan sistem pihak ketiga yang terintegrasi.
4. Konfigurasikan alat untuk mengakses lokasi terstandardisasi Anda
   a. Konfigurasikan alat-alat seperti Amazon Athena, QuickSight, atau solusi pihak ketiga untuk memiliki akses yang diperlukan ke lokasi terstandardisasi Anda. Konfigurasikan alat-alat ini agar dapat beroperasi di luar akun Alat Keamanan dengan akses baca lintas akun ke akun Arsip Log, jika diperlukan. Buat pelanggan di Amazon Security Lake untuk memberi alat-alat ini akses ke data Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP01 Memisahkan beban kerja menggunakan akun
- SEC07-BP04 Menentukan manajemen siklus hidup data
- SEC08-BP04 Menerapkan kontrol akses
- OPS08-BP02 Menganalisis log beban kerja

Dokumen terkait:

- Laporan resmi AWS: Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun
- Panduan Preskriptif AWS: Arsitektur Referensi Keamanan AWS (AWS SRA)
- Panduan Preskriptif AWS: Panduan pencatatan log dan pemantauan untuk pemilik aplikasi

Contoh terkait:

- Menggabungkan, mencari, dan memvisualisasikan data log dari sumber terdistribusi dengan Amazon Athena dan QuickSight
- Cara memvisualisasikan temuan Amazon Security Lake dengan QuickSight
- Buatlah wawasan yang didukung AI untuk Amazon Security Lake menggunakan Amazon SageMaker AI Studio dan Amazon Bedrock
- Identifikasi anomali keamanan siber dalam data Amazon Security Lake Anda menggunakan Amazon SageMaker AI
- Serap, transformasi, dan kirimkan peristiwa yang diterbitkan oleh Amazon Security Lake ke Amazon OpenSearch Service
- Sederhanakan analisis log AWS CloudTrail dengan pembuatan kueri bahasa alami di CloudTrail Lake

Alat terkait:

- Amazon Security Lake
- Integrasi Partner Amazon Security Lake
- Kerangka Kerja Skema Keamanan Siber Terbuka (OCSF)

- Amazon Athena
- QuickSight
- Amazon Bedrock

SEC04-BP03 Korelasikan dan perkaya data peringatan keamanan

Aktivitas tak terduga dapat menghasilkan beberapa peringatan keamanan yang dibuat oleh sumber yang berbeda, yang membutuhkan korelasi dan pengayaan lebih lanjut untuk memahami konteksnya secara lengkap. Menerapkan korelasi otomatis dan pengayaan peringatan keamanan untuk membantu mencapai identifikasi dan respons insiden yang lebih akurat.

Hasil yang diinginkan: Karena aktivitas menghasilkan peringatan yang berbeda dalam beban kerja dan lingkungan Anda, maka mekanisme otomatis menghubungkan data dan memperkaya data tersebut dengan informasi tambahan. Langkah-langkah sebelum pemrosesan (pre-processing) ini menyajikan pemahaman yang lebih mendetail tentang peristiwa, sehingga akan membantu penyelidik Anda dalam menentukan tingkat kekritisan peristiwa tersebut dan apakah peristiwa tersebut merupakan insiden yang memerlukan respons formal. Proses ini akan mengurangi beban pada tim pemantauan dan investigasi Anda.

Anti-pola umum:

- Grup orang yang berbeda-beda menyelidiki temuan dan peringatan yang dihasilkan oleh berbagai sistem, kecuali jika ditentukan lain berdasarkan persyaratan pemisahan tugas.
- Organisasi Anda menyalurkan semua data temuan dan peringatan keamanan ke lokasi terstandardisasi, tetapi mengharuskan penyelidik melakukan korelasi dan pengayaan data secara manual.
- Anda hanya mengandalkan intelijen sistem deteksi ancaman untuk melaporkan temuan dan menetapkan tingkat kekritisan.

Manfaat menjalankan praktik terbaik ini: Korelasi otomatis dan pengayaan peringatan akan membantu Anda dalam mengurangi beban kognitif secara keseluruhan dan persiapan data manual yang diperlukan peneliti Anda. Praktik ini dapat mengurangi waktu yang dibutuhkan untuk menentukan apakah peristiwa tersebut merepresentasikan sebuah insiden dan memulai respons formal. Konteks tambahan juga akan membantu Anda untuk menilai secara akurat tingkat keparahan yang sebenarnya dari suatu peristiwa, karena hal itu bisa jadi lebih tinggi atau lebih rendah dari yang diindikasikan oleh satu peringatan mana pun.

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

Panduan implementasi

Peringatan keamanan dapat berasal dari berbagai sumber yang ada di AWS, termasuk:

- Layanan-layanan seperti Amazon GuardDuty, AWS Security Hub, Amazon Macie, Amazon Inspector, AWS Config, AWS Identity and Access Management Access Analyzer, dan Penganalisis Akses Jaringan
- Peringatan dari analisis otomatis layanan AWS, infrastruktur, dan log aplikasi, seperti dari Security Analytics untuk Amazon OpenSearch Service.
- Alarm sebagai respons terhadap perubahan aktivitas penagihan Anda dari sumber seperti Amazon CloudWatch, Amazon EventBridge, atau AWS Budgets.
- Sumber pihak ketiga seperti umpan intelijen ancaman dan Solusi Mitra Keamanan dari AWS Partner Network
- Kontak oleh AWS Trust & Safety atau sumber lain, seperti pelanggan atau karyawan internal.

Dalam bentuknya yang paling mendasar, peringatan berisi informasi tentang siapa (principal atau identitas) yang melakukan apa (tindakan yang diambil) terhadap apa (sumber daya yang terdampak tindakan). Untuk masing-masing sumber ini, identifikasi apakah Anda dapat membuat pemetaan di seluruh pengidentifikasi untuk identitas, tindakan, dan sumber daya ini sebagai landasan untuk melakukan korelasi. Hal ini dapat berupa mengintegrasikan sumber peringatan dengan alat manajemen informasi dan peristiwa keamanan (SIEM) untuk melakukan korelasi secara otomatis untuk Anda, membuat pipeline dan pemrosesan data Anda sendiri, atau kombinasi keduanya.

Contoh layanan yang dapat melakukan korelasi untuk Anda adalah Amazon Detective. Amazon Detective melakukan penyerapan peringatan yang berkelanjutan dari berbagai sumber AWS dan pihak ketiga, serta menggunakan berbagai bentuk intelijen untuk menyusun grafik visual tentang hubungan peringatan-peringatan tersebut untuk membantu penyelidikan.

Meskipun tingkat kekritisan awal sebuah peringatan dapat membantu dalam menentukan prioritisas, konteks yang mendasari terjadinya peringatan tersebut akan menentukan tingkat kekritisan yang sebenarnya. Sebagai contoh, Amazon GuardDuty dapat mengingatkan bahwa instans Amazon EC2 dalam beban kerja Anda sedang mengueri nama domain yang tidak terduga. GuardDuty mungkin menetapkan tingkat kekritisan yang rendah untuk peringatan ini sendiri. Namun, korelasi otomatis dengan aktivitas lainnya yang terjadi kira-kira pada saat peringatan diberikan mungkin menunjukkan bahwa beberapa ratus instans EC2 di-deploy dengan identitas yang sama, sehingga

akan meningkatkan biaya operasi secara keseluruhan. Dalam peristiwa ini, konteks peristiwa yang berkorelasi ini mungkin memerlukan peringatan keamanan baru dan tingkat kekritisannya mungkin harus diubah ke tinggi, sehingga akan mempercepat tindakan lebih lanjut.

Langkah-langkah implementasi

1. Identifikasi sumber untuk informasi peringatan keamanan. Pahami cara peringatan dari sistem ini merepresentasikan identitas, tindakan, dan sumber daya untuk menentukan di mana korelasi mungkin dilakukan.
2. Tetapkan sebuah mekanisme untuk merekam peringatan dari sumber yang berbeda-beda. Pertimbangkan layanan-layanan seperti Security Hub, EventBridge, dan CloudWatch untuk tujuan ini.
3. Identifikasi sumber untuk korelasi dan pengayaan data. Contoh sumber meliputi AWS CloudTrail, Log Aliran VPC, log Route 53 Resolver, serta log infrastruktur dan aplikasi. Salah satu atau semua log ini dapat dikonsumsi melalui integrasi tunggal dengan Amazon Security Lake.
4. Integrasikan peringatan Anda dengan sumber korelasi dan pengayaan data untuk membuat konteks peristiwa keamanan yang lebih mendetail dan menetapkan tingkat kekritisannya.
    a. Amazon Detective, alat SIEM, atau solusi pihak ketiga lainnya dapat melakukan penyerapan, korelasi, dan pengayaan data pada tingkat tertentu secara otomatis.
    b. Anda juga dapat menggunakan layanan-layanan AWS untuk membuat solusi Anda sendiri. Misalnya, Anda dapat menginvokasi fungsi AWS Lambda untuk menjalankan kueri Amazon Athena terhadap AWS CloudTrail atau Amazon Security Lake, dan memublikasikan hasilnya ke EventBridge.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC10-BP03 Menyiapkan kemampuan forensik
- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti
- REL06-BP03 Mengirimkan notifikasi (Pemrosesan dan pembuatan alarm waktu nyata)

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS

Contoh terkait:

- Cara memperkaya temuan AWS Security Hub dengan metadata akun

Alat terkait:

- Amazon Detective
- Amazon EventBridge
- AWS Lambda
- Amazon Athena

SEC04-BP04 Mulai melakukan remediasi untuk sumber daya yang tidak mematuhi persyaratan

Kontrol deteksi Anda mungkin memberikan peringatan tentang sumber daya yang tidak mematuhi persyaratan-persyaratan konfigurasi Anda. Anda dapat mulai melakukan remediasi yang ditentukan secara programatis, baik secara manual maupun otomatis, untuk melakukan remediasi terhadap berbagai sumber daya ini dan membantu meminimalkan dampak-dampak yang mungkin ditimbulkannya. Ketika Anda menentukan remediasi secara programatis, Anda dapat mengambil tindakan yang cepat dan konsisten.

Meskipun otomatisasi dapat meningkatkan operasi keamanan, Anda harus mengimplementasikan dan mengelola otomatisasi dengan hati-hati. Terapkan mekanisme pengawasan dan kontrol yang tepat untuk memastikan bahwa respons otomatis berjalan efektif, akurat, dan selaras dengan kebijakan organisasi dan tingkat risiko yang dapat diterima.

Hasil yang diinginkan: Anda menentukan standar-standar konfigurasi sumber daya bersama dengan langkah-langkah untuk memulihkan ketika sumber daya terdeteksi tidak sesuai. Jika memungkinkan, Anda sudah harus menentukan remediasi secara programatis sehingga remediasi ini dapat dimulai baik secara manual maupun dengan otomatisasi. Sistem deteksi tersedia untuk mengidentifikasi sumber daya yang tidak mematuhi persyaratan dan memublikasikan peringatan ke alat-alat tersentralisasi yang dipantau oleh personel keamanan Anda. Dengan alat-alat ini, remediasi programatis Anda dapat dijalankan, baik secara manual maupun otomatis. Remediasi otomatis memiliki mekanisme pengawasan dan kontrol yang tepat untuk mengatur penggunaannya.

Anti-pola umum:

- Anda mengimplementasikan otomatisasi, tetapi gagal menguji dan memvalidasi tindakan-tindakan remediasi secara menyeluruh. Hal ini dapat menimbulkan konsekuensi-konsekuensi yang tidak diinginkan, seperti mengganggu operasi bisnis yang sah atau menyebabkan ketidakstabilan terhadap sistem.
- Anda mengoptimalkan waktu dan prosedur respons melalui otomatisasi, tetapi tanpa pemantauan dan mekanisme yang tepat yang memungkinkan intervensi dan penilaian manusia saat diperlukan.
- Anda hanya mengandalkan remediasi, bukannya menggunakan remediasi sebagai salah satu bagian dari program respons dan pemulihan insiden yang lebih luas.

Manfaat menjalankan praktik terbaik ini: Remediasi otomatis dapat merespons kesalahan konfigurasi lebih cepat daripada proses manual, yang akan membantu Anda meminimalkan potensi dampak bisnis dan mengurangi jendela peluang terjadinya penggunaan yang tidak diinginkan. Ketika Anda menentukan remediasi secara terprogram, remediasi akan diterapkan secara konsisten, dan hal ini akan mengurangi risiko kesalahan manusia. Otomatisasi juga dapat menangani volume peringatan yang lebih besar yang muncul secara bersamaan, yang merupakan hal yang sangat penting di lingkungan yang beroperasi dalam skala besar.

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

Panduan implementasi

Seperti dijelaskan dalam SEC01-BP03 Identifikasikan dan validasikan tujuan kontrol, layanan seperti AWS Config dan AWS Security Hub dapat membantu Anda memantau konfigurasi sumber daya di akun Anda untuk kepatuhan terhadap persyaratan Anda. Ketika sumber daya yang tidak mematuhi persyaratan terdeteksi, layanan seperti AWS Security Hub dapat membantu merutekan peringatan dengan tepat dan melakukan remediasi. Solusi ini akan menyediakan sebuah tempat terpusat bagi penyelidik keamanan Anda untuk memantau masalah dan mengambil tindakan korektif.

Meskipun beberapa situasi sumber daya yang tidak mematuhi persyaratan bersifat unik dan memerlukan penilaian manusia untuk diremediasi, namun situasi lainnya memiliki respons standar yang dapat Anda tentukan secara programatis. Misalnya, respons standar terhadap grup keamanan VPC yang mengalami salah konfigurasi dapat berupa tindakan menghapus aturan yang tidak diizinkan dan memberi tahu pemiliknya. Respons dapat didefinisikan dalam fungsi AWS Lambda, dokumen AWS Systems Manager Automation, atau melalui lingkungan kode lain yang Anda inginkan. Pastikan lingkungan tersebut dapat mengautentikasi ke AWS menggunakan peran IAM dengan jumlah izin minimal yang diperlukan untuk mengambil tindakan korektif.

Setelah Anda menentukan perbaikan yang diinginkan, Anda kemudian dapat menentukan cara pilihan yang akan Anda gunakan untuk memulainya. AWS Config dapat memulai perbaikan untuk Anda. Jika Anda menggunakan Security Hub, Anda dapat melakukannya melalui tindakan kustom, yang menerbitkan informasi temuan ke Amazon EventBridge. Aturan EventBridge kemudian dapat memulai remediasi Anda. Anda dapat mengonfigurasi remediasi di Security Hub untuk dijalankan secara otomatis atau manual.

Untuk remediasi terprogram, sebaiknya Anda memiliki log dan audit komprehensif untuk tindakan yang diambil, serta hasilnya. Lakukan peninjauan dan analisis terhadap log ini untuk menilai efektivitas proses yang terjadi secara otomatis, dan mengidentifikasi area perbaikan yang mungkin perlu dilakukan. Rekam log di Log Amazon CloudWatch dan hasil remediasi sebagai catatan temuan di Security Hub.

Sebagai titik awal, pertimbangkan Respons Keamanan Otomatis AWS, yang memiliki remediasi pra-bangun untuk menyelesaikan kesalahan konfigurasi keamanan yang sering terjadi.

Langkah-langkah implementasi

1. Analisis dan prioritaskan peringatan.
    a. Konsolidasikan peringatan keamanan dari berbagai layanan AWS ke dalam Security Hub untuk visibilitas, prioritas, dan remediasi terpusat.
2. Kembangkan remediasi.
    a. Gunakan layanan-layanan seperti Systems Manager dan AWS Lambda untuk menjalankan remediasi programatis.
3. Konfigurasikan cara remediasi dimulai.
    a. Menggunakan Systems Manager, tentukan tindakan kustom yang memublikasikan temuan ke EventBridge. Konfigurasikan tindakan-tindakan ini untuk dimulai secara manual atau otomatis.
    b. Anda juga dapat menggunakan Amazon Simple Notification Service (SNS) untuk mengirim notifikasi dan peringatan kepada para pemangku kepentingan terkait (seperti tim keamanan atau tim respons insiden) untuk intervensi atau eskalasi manual, jika diperlukan.
4. Tinjau dan analisis log remediasi untuk menentukan efektivitas dan peningkatan.
    a. Mengirimkan output log ke CloudWatch Logs. Rekam hasil sebagai catatan temuan di Security Hub.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC06-BP03 Kurangi manajemen manual dan akses interaktif

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS - Deteksi

Contoh terkait:

- Respons Keamanan Otomatis di AWS
- Pantau pasangan kunci instans EC2 dengan menggunakan AWS Config
- Buat aturan kustom AWS Config dengan menggunakan kebijakan AWS CloudFormation Guard
- Secara otomatis memulihkan instans dan klaster Amazon RDS DB yang tidak terenkripsi

Alat terkait:

- AWS Systems Manager Automation
- Respons Keamanan Otomatis di AWS

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 dari sebuah program keamanan informasi. Fungsinya adalah untuk memastikan sistem dan layanan dalam beban kerja Anda terlindungi dari potensi kerentanan 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 melakukan hardening, 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, Zona Lokal AWS, dan Outposts AWS

Pastikan Anda sudah familiar dengan Wilayah, Zona Ketersediaan, Zona Lokal AWS, dan Outposts AWS, yang merupakan komponen infrastruktur global yang aman dari AWS.

AWS memiliki konsep Wilayah, yakni lokasi fisik di seluruh dunia tempat kami membuat klaster-klaster pusat-pusat data. Kami menyebut setiap kelompok pusat data ini dengan sebutan Zona Ketersediaan (AZ). Setiap Wilayah AWS terdiri dari beberapa AZ yang terisolasi dan dipisahkan secara fisik di sebuah area geografis. Jika Anda memiliki persyaratan residensi data, Anda dapat memilih Wilayah AWS yang dekat dengan lokasi yang Anda kehendaki. Anda mempertahankan dan 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 yang independen. Jika suatu aplikasi dipartisi secara lintas AZ, Anda akan lebih terisolasi dan terlindungi dari permasalahan-permasalahan yang mungkin dihadapi, misalnya 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 yang lain, meskipun semuanya berada dalam jarak 100 km (60 mil) dari satu sama lain. Semua AZ di sebuah Wilayah AWS saling terhubung dengan bandwidth yang tinggi, jaringan latensi yang rendah, menggunakan serat metro khusus yang sepenuhnya redundan, yang dapat menyediakan jaringan throughput yang tinggi dan latensi yang 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. AWS Wilayah memenuhi tingkat keamanan, kepatuhan, dan perlindungan data tertinggi.

Zona Lokal AWS menempatkan layanan komputasi, penyimpanan, basis data, dan layanan-layanan AWS terpilih lainnya lebih dekat dengan pengguna akhir. Dengan Zona Lokal AWS, Anda dapat dengan mudah menjalankan aplikasi yang sangat berat yang memerlukan latensi satu digit milidetik ke pengguna akhir, seperti pembuatan media dan konten hiburan, gaming waktu nyata, simulasi waduk, 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. AWS Zona Lokal menyediakan koneksi dengan bandwidth yang tinggi dan aman di 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 native AWS untuk melakukan virtualisasi terhadap semua pusat data, ruang kolokasi, 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, ada beberapa pendekatan perlindungan infrastruktur. Bagian-bagian berikutnya akan menjelaskan cara menggunakan pendekatan-pendekatan ini.

Topik

Melindungi jaringan

Pengguna, baik dari tenaga kerja maupun pelanggan Anda, bisa berlokasi di mana saja. Anda perlu beralih dari model tradisional yang memberikan kepercayaan untuk 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 dengan pendekatan Zero Trust adalah model di mana komponen-komponen aplikasi atau layanan mikro dianggap terpisah satu sama lain dan komponen-komponen atau layanan mikro tersebut tidak saling mempercayai satu sama lain.

Perencanaan dan manajemen yang cermat pada desain jaringan Anda akan membentuk fondasi bagi Anda untuk menyediakan pemisahan dan batasan untuk sumber daya yang ada dalam beban kerja Anda. Karena banyak sumber daya di beban kerja Anda beroperasi dalam sebuah VPC dan mewarisi properti-properti keamanan, Anda harus membuat desain yang didukung dengan mekanisme pengawasan dan perlindungan yang diperkuat oleh otomatisasi. Demikian juga, untuk beban kerja yang beroperasi di luar sebuah VPC, dengan menggunakan layanan murni edge dan/atau nirserver, praktik-praktik terbaik tersebut berlaku dalam pendekatan yang lebih sederhana. Lihat Lensa Aplikasi Nirserver AWS Well-Architected untuk panduan khusus tentang keamanan nirserver.

Praktik terbaik

- SEC05-BP01 Buat lapisan jaringan
- SEC05-BP02 Kontrol arus lalu lintas dalam lapisan jaringan Anda
- SEC05-BP03 Menerapkan perlindungan berbasis inspeksi
- SEC05-BP04 Mengotomatiskan perlindungan jaringan

SEC05-BP01 Buat lapisan jaringan

Segmentasikan topologi jaringan Anda ke dalam lapisan yang berbeda-beda berdasarkan pengelompokan logis komponen beban kerja Anda sesuai dengan sensitivitas data dan persyaratan aksesnya. Bedakan antara komponen yang memerlukan akses masuk dari internet, seperti titik akhir web publik, dan komponen yang hanya membutuhkan akses internal, seperti basis data.

Hasil yang diinginkan: Lapisan jaringan Anda adalah bagian dari defense-in-depth pendekatan integral terhadap keamanan yang melengkapi otentikasi identitas dan strategi otorisasi beban kerja Anda. Lapisan-lapisan diterapkan berdasarkan sensitivitas data dan persyaratan akses, dengan arus lalu lintas dan mekanisme kontrol yang sesuai.

Anti-pola umum:

- Anda membuat semua sumber daya dalam satu VPC atau subnet.
- Anda menyusun konsep lapisan-lapisan jaringan tanpa mempertimbangkan persyaratan sensitivitas data, perilaku komponen, atau fungsionalitas.
- Anda menggunakan VPCs dan subnet sebagai default untuk semua pertimbangan lapisan jaringan, dan Anda tidak mempertimbangkan bagaimana layanan AWS terkelola memengaruhi topologi Anda.

Manfaat menerapkan praktik terbaik ini: Membangun lapisan jaringan adalah langkah pertama dalam membatasi jalur-jalur yang tidak perlu melalui jaringan, terutama jalur-jalur yang mengarah pada sistem dan data kritis. Hal ini akan mempersulit pelaku yang tidak sah untuk mendapatkan akses ke jaringan Anda dan menavigasi ke sumber daya tambahan di dalamnya. Lapisan-lapisan jaringan terpisah (discrete) bermanfaat dalam mengurangi cakupan analisis untuk sistem inspeksi, seperti untuk deteksi intrusi atau pencegahan malware. Hal ini dapat mengurangi potensi positif palsu dan overhead pemrosesan yang tidak perlu.

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

Panduan implementasi

Saat merancang arsitektur beban kerja, biasanya berbagai komponen dipisahkan menjadi lapisan yang berbeda-beda berdasarkan tanggung jawabnya. Misalnya, sebuah aplikasi web dapat memiliki lapisan presentasi, lapisan aplikasi, dan lapisan data. Anda dapat mengambil pendekatan yang serupa saat merancang desain topologi jaringan Anda. Kontrol jaringan yang mendasarinya dapat membantu Anda memberlakukan persyaratan akses data pada beban kerja Anda. Misalnya, dalam arsitektur aplikasi web tiga tingkat, Anda dapat menyimpan file lapisan presentasi statis Anda di Amazon S3 dan menyajikannya dari jaringan pengiriman konten CDN (), seperti Amazon. CloudFront Lapisan aplikasi dapat memiliki titik akhir publik yang berfungsi Application Load Balancer ALB () di subnet publik VPCAmazon (mirip dengan zona demiliterisasi, DMZ atau), dengan layanan back-end yang digunakan ke subnet pribadi. Lapisan data tersebut, yang merupakan sumber daya hosting seperti basis data dan sistem file bersama, dapat berada di subnet privat yang berbeda dari sumber daya lapisan aplikasi Anda. Pada setiap batas lapisan ini (CDN, subnet publik, subnet pribadi), Anda dapat menerapkan kontrol yang memungkinkan hanya lalu lintas resmi untuk melintasi batas-batas tersebut.

Serupa dengan pemodelan lapisan jaringan berdasarkan tujuan fungsional komponen beban kerja Anda, pertimbangkan juga sensitivitas data yang diproses. Menggunakan contoh aplikasi web tersebut, meskipun semua layanan beban kerja Anda mungkin berada dalam lapisan aplikasi, layanan yang berbeda dapat memproses data dengan tingkat sensitivitas yang berbeda. Dalam hal ini, membagi lapisan aplikasi menggunakan beberapa subnet pribadi, berbeda VPCs dalam hal yang sama Akun AWS, atau bahkan berbeda VPCs berbeda Akun AWS untuk setiap tingkat sensitivitas data mungkin sesuai dengan persyaratan kontrol Anda.

Pertimbangan lebih lanjut yang harus dilakukan untuk lapisan jaringan adalah konsistensi perilaku dari komponen-komponen yang dimiliki oleh beban kerja Anda. Melanjutkan dengan contoh ini, di lapisan aplikasi, Anda mungkin memiliki layanan yang menerima input dari pengguna akhir atau integrasi sistem eksternal yang secara inheren lebih berisiko daripada input ke layanan lain.

Contohnya termasuk unggahan file, skrip kode yang dijalankan, pemindaian email, dan sebagainya. Dengan menempatkan layanan-layanan tersebut di lapisan jaringannya sendiri, akan terbentuk batas isolasi yang lebih kuat di sekitarnya, dan perilaku uniknya dapat dicegah agar tidak menghasilkan peringatan positif palsu dalam sistem inspeksi.

Sebagai bagian dari desain Anda, pertimbangkan bagaimana menggunakan layanan AWS terkelola memengaruhi topologi jaringan Anda. Jelajahi bagaimana layanan seperti Amazon VPC Lattice dapat membantu mempermudah interoperabilitas komponen beban kerja Anda di seluruh lapisan jaringan. Saat menggunakan AWS Lambda, terapkan di VPC subnet Anda kecuali ada alasan khusus untuk tidak melakukannya. Tentukan di mana VPC titik akhir dan AWS PrivateLinkdapat menyederhanakan mematuhi kebijakan keamanan yang membatasi akses ke gateway internet.

Langkah-langkah implementasi

1. Tinjau arsitektur beban kerja Anda. Kelompokkan komponen dan layanan secara logis berdasarkan fungsi yang dijalankan, sensitivitas data yang diproses, dan perilakunya.
2. Untuk komponen-komponen yang merespons permintaan dari internet, pertimbangkan untuk menggunakan penyeimbang beban atau perantara lainnya guna menyediakan titik akhir publik. Jelajahi pengalihan kontrol keamanan dengan menggunakan layanan terkelola, seperti Amazon API Gateway CloudFront, Elastic Load Balancing, AWS Amplifydan untuk menampung titik akhir publik.
3. Untuk komponen yang berjalan di lingkungan komputasi, seperti EC2 instans Amazon, AWS Fargatecontainer, atau fungsi Lambda, terapkan ini ke subnet pribadi berdasarkan grup Anda dari langkah pertama.
4. Untuk AWS layanan yang dikelola sepenuhnya, seperti Amazon DynamoDB, Amazon Kinesis, atau SQSAmazon, pertimbangkan untuk VPC menggunakan titik akhir sebagai default untuk akses melalui alamat IP pribadi.

Sumber daya

Praktik-praktik terbaik terkait:

- REL02 Rencanakan topologi jaringan Anda
- PERF04-BP01 Memahami bagaimana jaringan memengaruhi kinerja

Video terkait:

- AWS re: invent 2023 - yayasan jaringan AWS

Contoh terkait:

- VPCcontoh
- Akses aplikasi kontainer secara pribadi di Amazon ECS dengan menggunakan AWS Fargate, AWS PrivateLink, dan Network Load Balancer
- Sajikan konten statis dalam bucket Amazon S3 melalui VPC dengan menggunakan Amazon CloudFront

SEC05-BP02 Kontrol arus lalu lintas dalam lapisan jaringan Anda

Dalam lapisan-lapisan jaringan Anda, gunakan segmentasi lebih lanjut untuk membatasi lalu lintas hanya ke arus yang diperlukan untuk masing-masing beban kerja. Pertama, fokus pada pengendalian lalu lintas antara internet atau sistem eksternal lainnya ke beban kerja dan lingkungan Anda (lalu lintas utara-selatan). Setelah itu, lihat arus yang terjadi antara komponen dan sistem yang berbeda (lalu lintas timur-barat).

Hasil yang diinginkan: Anda hanya mengizinkan arus jaringan yang diperlukan untuk komponen beban kerja Anda untuk melakukan komunikasi satu sama lain dan dan klien mereka dan layanan lain yang mereka andalkan. Desain Anda mempertimbangkan hal-hal seperti lalu lintas masuk dan keluar publik dibandingkan dengan privat, klasifikasi data, peraturan regional, dan persyaratan protokol. Jika memungkinkan, Anda menyukai arus point-to-point melalui peering jaringan sebagai bagian dari prinsip desain hak akses paling rendah.

Anti-pola umum:

- Anda mengambil pendekatan berbasis perimeter untuk keamanan jaringan dan hanya mengontrol arus lalu lintas di batas lapisan-lapisan jaringan Anda.
- Anda menganggap semua lalu lintas yang ada dalam sebuah lapisan jaringan sudah diautentikasi dan diotorisasi.
- Anda dapat menerapkan kontrol untuk salah satu lalu lintas masuk atau lalu lintas keluar, tetapi tidak dapat menerapkan untuk keduanya.
- Anda hanya mengandalkan komponen-komponen beban kerja dan kontrol jaringan untuk melakukan autentikasi dan meberikan otorisasi terhadap lalu lintas.

Manfaat menerapkan praktik terbaik ini: Praktik ini akan membantu Anda dalam mengurangi risiko pergerakan yang tidak sah dalam jaringan Anda dan menambahkan lapisan otorisasi tambahan ke beban kerja Anda. Dengan melakukan kontrol terhadap arus lalu lintas, Anda dapat membatasi cakupan dampak insiden keamanan serta mempercepat deteksi dan respons.

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

Panduan implementasi

Meskipun lapisan jaringan membantu menetapkan batas-batas di sekitar komponen beban kerja Anda yang memberikan fungsi, tingkat sensitivitas data, dan perilaku yang serupa, Anda dapat membuat tingkat kontrol lalu lintas yang jauh lebih terperinci menggunakan berbagai teknik untuk membagi komponen lebih lanjut dalam lapisan tersebut yang mengikuti prinsip hak akses paling rendah. Dalam AWS, lapisan jaringan sebagian besar ditentukan dengan menggunakan subnet sesuai dengan rentang alamat IP yang ada dalam Amazon VPC. Lapisan juga dapat ditentukan menggunakan VPC yang berbeda-beda, seperti untuk melakukan pengelompokan lingkungan layanan mikro berdasarkan domain bisnis. Saat menggunakan banyak VPC, mediasi perutean menggunakan AWS Transit Gateway. Meskipun ini memberikan kontrol lalu lintas pada tingkat Layer 4 (alamat IP dan rentang port) menggunakan grup keamanan dan tabel rute, Anda dapat memperoleh kontrol lebih lanjut dengan menggunakan layanan tambahan, seperti AWS PrivateLink, Amazon Route 53 Resolver DNS Firewall, AWS Network Firewall dan AWS WAF.

Memahami dan menginventarisasi aliran data dan persyaratan komunikasi beban kerja Anda dalam hal pihak yang memulai koneksi, port, protokol, dan lapisan jaringan. Lakukan evaluasi terhadap protokol yang tersedia untuk membangun koneksi dan mentransmisikan data untuk memilih protokol yang memenuhi persyaratan perlindungan Anda (misalnya, HTTPS, bukan HTTP). Terapkan persyaratan-persyaratan ini di batas-batas jaringan Anda dan dalam masing-masing lapisan. Setelah persyaratan-persyaratan ini diidentifikasi, tinjau opsi-opsi untuk mengizinkan lalu lintas yang diperlukan saja yang dapat mengalir di setiap titik koneksi. Titik awal yang baik adalah menggunakan grup keamanan dalam VPC Anda, karena mereka dapat dilampirkan ke sumber daya yang menggunakan Antarmuka Jaringan Elastis (ENI), seperti instans Amazon EC2, tugas Amazon ECS, pod Amazon EKS, atau basi data Amazon RDS. Tidak seperti firewall Lapisan 4, sebuah grup keamanan dapat memiliki aturan yang mengizinkan lalu lintas dari grup keamanan lain berdasarkan pengidentifikasinya, sehingga hal itu akan meminimalkan pembaruan seiring berubahnya sumber daya dalam grup dari waktu ke waktu. Anda juga dapat memfilter lalu lintas dengan aturan masuk dan keluar dengan menggunakan grup keamanan.

Ketika lalu lintas bergerak di antara VPC, biasanya peering VPC digunakan untuk perutean sederhana atau AWS Transit Gateway untuk perutean yang kompleks. Dengan pendekatan ini,

Anda memfasilitasi arus lalu lintas yang terjadi antara rentang alamat IP jaringan sumber dan tujuan. Namun demikian, jika beban kerja Anda hanya memerlukan arus lalu lintas antara komponen tertentu di VPC yang berbeda, sebaiknya Anda menggunakan koneksi point-to-point dengan menggunakan AWS PrivateLink. Untuk melakukan hal ini, identifikasi layanan mana yang harus bertindak sebagai produsen dan layanan mana yang harus bertindak sebagai konsumen. Lakukan deployment penyeimbang beban yang kompatibel untuk produsen, nyalakan PrivateLink sesuai dengan itu, dan kemudian terima permintaan koneksi oleh konsumen. Layanan produsen kemudian diberi alamat IP privat dari VPC konsumen yang dapat digunakan konsumen untuk membuat permintaan berikutnya. Pendekatan ini mengurangi kebutuhan untuk melakukan peering jaringan. Sertakan biaya untuk pemrosesan data dan penyeimbangan beban sebagai bagian dari evaluasi PrivateLink.

Meskipun grup keamanan dan PrivateLink dapt membantu mengontrol alur yang ada di antara komponen beban kerja Anda, hal lain yang juga harus dipertimbangkan adalah bagaimana mengontrol domain DNS mana yang diizinkan untuk diakses oleh sumber daya Anda (jika ada). Bergantung pada konfigurasi DHCP VPC Anda, Anda dapat mempertimbangkan dua layanan AWS berbeda untuk tujuan ini. Sebagian besar pelanggan menggunakan layanan DNS Route 53 Resolver default (juga disebut server DNS Amazon atau AmazonProvidedDNS) yang tersedia untuk VPC di alamat +2 dari rentang CIDR-nya. Dengan pendekatan ini, Anda dapat membuat aturan Firewall DNS dan kemudian mengaitkan aturan tersebut ke VPC Anda yang menentukan tindakan apa yang harus diambil untuk daftar domain yang Anda berikan.

Jika Anda tidak menggunakan Route 53 Resolver, atau jika Anda ingin melengkapi Resolver dengan kemampuan inspeksi dan kontrol aliran yang lebih mendalam selain pemfilteran domain, pertimbangkan untuk melakukan deployment AWS Network Firewall. Layanan ini memeriksa masing-masing paket individual dengan menggunakan aturan stateless atau stateful untuk menentukan apakah akan menolak atau mengizinkan lalu lintas. Anda dapat mengambil pendekatan serupa untuk memfilter lalu lintas web masuk ke titik akhir publik Anda dengan menggunakan AWS WAF. Untuk membaca panduan lebih lanjut tentang layanan ini, lihat SEC05-BP03 Menerapkan perlindungan berbasis inspeksi.

Langkah-langkah implementasi

1. Identifikasi aliran data yang diperlukan antara komponen-komponen beban kerja Anda.
2. Terapkan beberapa kontrol dengan pendekatan pertahanan mendalam untuk lalu lintas masuk dan keluar, termasuk penggunaan grup keamanan, dan tabel rute.
3. Gunakan firewall untuk menentukan kontrol terperinci terhadap lalu lintas jaringan masuk, keluar, dan di seluruh VPC Anda, seperti Firewall DNS Route 53 Resolver, AWS Network Firewall, dan

AWS WAF. Pertimbangkan untuk menggunakan AWS Firewall Manager untuk mengonfigurasi dan mengelola aturan firewall secara terpusat di seluruh organisasi Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- REL03-BP01 Memilih cara untuk menyegmentasi beban kerja
- SEC09-BP02 Menerapkan enkripsi data bergerak

Dokumen terkait:

- Praktik terbaik keamanan untuk VPC Anda
- Tips Optimalisasi Jaringan AWS
- Panduan untuk Keamanan Jaringan di AWS
- Amankan lalu lintas jaringan keluar VPC Anda di AWS Cloud

Alat terkait:

- AWS Firewall Manager

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 2023: Firewall dan tempat meletakkannya

SEC05-BP03 Menerapkan perlindungan berbasis inspeksi

Siapkan titik inspeksi lalu lintas di antara lapisan jaringan Anda untuk memastikan data bergerak cocok dengan kategori dan pola yang diharapkan. Lakukan analisis terhadap arus lalu lintas, metadata, dan pola untuk membantu Anda mengidentifikasi, mendeteksi, dan merespons peristiwa dengan lebih efektif.

Hasil yang diinginkan: Lalu lintas yang melintas antara lapisan jaringan Anda diperiksa dan diberi otorisasi. Keputusan izinkan (allow) dan tolak (deny) didasarkan pada aturan-aturan eksplisit,

intelijen ancaman, dan penyimpangan dari perilaku acuan dasar. Perlindungan menjadi lebih ketat ketika lalu lintas makin mendekati data sensitif.

Anti-pola umum:

- Hanya mengandalkan aturan-aturan firewall berdasarkan port dan protokol. Tidak memanfaatkan sistem cerdas.
- Menulis aturan firewall berdasarkan pola ancaman spesifik saat ini yang masih dapat berubah.
- Hanya memeriksa lalu lintas yang lalu lintasnya bergerak dari subnet privat ke publik, atau dari subnet publik ke Internet.
- Tidak memiliki gambaran acuan dasar tentang lalu lintas jaringan Anda untuk dijadikan pembanding dengan anomali perilaku.

Manfaat menerapkan praktik terbaik ini: Sistem inspeksi akan memungkinkan Anda untuk membuat aturan cerdas, seperti mengizinkan atau menolak lalu lintas hanya ketika kondisi tertentu terjadi dalam data lalu lintas. Manfaatkan set aturan yang dikelola dari AWS dan mitra, berdasarkan intelijen ancaman terbaru, karena lanskap ancaman berubah seiring waktu. Hal ini akan menurunkan overhead pemeliharaan aturan dan pencarian indikator penyusupan, sehingga dapat mengurangi potensi positif palsu.

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

Panduan implementasi

Memiliki kontrol halus atas lalu lintas jaringan stateful dan stateless Anda menggunakan AWS Network Firewall, atau Firewall dan Intrusion Prevention Systems (IPS) lainnya yang AWS Marketplace dapat Anda gunakan di belakang Gateway Load Balancer (). GWLBAWS Network Firewall mendukung IPS spesifikasi open source yang kompatibel dengan Suricata untuk membantu melindungi beban kerja Anda.

Baik solusi AWS Network Firewall dan vendor yang menggunakan GWLB dukungan model penyebaran inspeksi inline yang berbeda. Misalnya, Anda dapat melakukan inspeksi per- VPC basis, memusatkan dalam inspeksiVPC, atau menerapkan dalam model hibrida di mana lalu lintas timur-barat mengalir melalui inspeksi VPC dan masuknya Internet diperiksa per-. VPC Pertimbangan lain adalah apakah solusi tersebut mendukung membuka bungkusan Transport Layer Security (TLS), memungkinkan inspeksi paket mendalam untuk arus lalu lintas yang dimulai di kedua arah. Untuk informasi selengkapnya dan detail mendalam tentang konfigurasi ini, lihat panduan Praktik Terbaik AWS Network Firewall.

Jika Anda menggunakan solusi yang melakukan out-of-band inspeksi, seperti analisis pcap data paket dari antarmuka jaringan yang beroperasi dalam mode promiscuous, Anda dapat mengonfigurasi mirroring lalu lintas. VPC Lalu lintas yang di-mirroring diperhitungkan terhadap bandwidth yang tersedia dari antarmuka Anda dan dikenai biaya transfer data yang sama dengan lalu lintas yang tidak di-mirroring. Anda dapat melihat apakah versi virtual dari peralatan ini tersedia di AWS Marketplace, yang dapat mendukung penyebaran sebaris di belakang file. GWLB

Untuk komponen yang bertransaksi melalui protokol HTTP berbasis, lindungi aplikasi Anda dari ancaman umum dengan firewall aplikasi web (). WAF AWS WAFadalah firewall aplikasi web yang memungkinkan Anda memantau dan memblokir HTTP permintaan yang sesuai dengan aturan yang dapat dikonfigurasi sebelum mengirim ke Amazon API Gateway, Amazon CloudFront, AWS AppSync atau Application Load Balancer. Pertimbangkan inspeksi paket mendalam ketika Anda mengevaluasi penerapan firewall aplikasi web Anda, karena beberapa mengharuskan Anda untuk menghentikan TLS sebelum inspeksi lalu lintas. Untuk memulai AWS WAF, Anda dapat menggunakan Peraturan yang Dikelola AWSkombinasi dengan milik Anda sendiri, atau menggunakan integrasi mitra yang ada.

Anda dapat mengelola AWS WAF, AWS Shield Advanced AWS Network Firewall, dan grup VPC keamanan Amazon secara terpusat di seluruh AWS Organisasi Anda. AWS Firewall Manager

Langkah-langkah implementasi

1. Tentukan apakah Anda dapat mencakup aturan inspeksi secara luas, seperti melalui inspeksiVPC, atau jika Anda memerlukan pendekatan per detail yang lebih terperinci. VPC
2. Untuk solusi-solusi inspeksi inline:
   a. Jika menggunakan AWS Network Firewall, buat aturan, kebijakan firewall, dan firewall itu sendiri. Setelah ini dikonfigurasi, Anda dapat merutekan lalu lintas ke titik akhir firewall untuk mengaktifkan inspeksi.
   b. Jika menggunakan alat pihak ketiga dengan Gateway Load Balancer (GWLB), gunakan dan konfigurasikan alat Anda di satu atau beberapa zona ketersediaan. Kemudian, buat, layanan titik akhirGWLB, titik akhir, dan konfigurasikan perutean untuk lalu lintas Anda.
3. Untuk solusi out-of-band inspeksi:
   1. Aktifkan Pencerminan VPC Lalu Lintas pada antarmuka tempat lalu lintas masuk dan keluar harus dicerminkan. Anda dapat menggunakan EventBridge aturan Amazon untuk menjalankan AWS Lambda fungsi guna mengaktifkan pencerminan lalu lintas pada antarmuka saat sumber daya baru dibuat. Arahkan sesi traffic mirroring ke Penyeimbang Beban Jaringan di depan alat Anda yang memproses lalu lintas.

4. Untuk solusi lalu lintas web masuk:
    a. Untuk mengkonfigurasi AWS WAF, mulailah dengan mengkonfigurasi daftar kontrol akses web (webACL). Web ACL adalah kumpulan aturan dengan tindakan default yang diproses secara serial (ALLOWatauDENY) yang menentukan bagaimana Anda WAF menangani lalu lintas. Anda dapat membuat aturan dan grup Anda sendiri atau menggunakan grup aturan AWS terkelola di web AndaACL.
    b. Setelah web Anda ACL dikonfigurasi, kaitkan web ACL dengan AWS sumber daya (seperti Application Load Balancer, API Gateway RESTAPI, atau CloudFront distribusi) untuk mulai melindungi lalu lintas web.

Sumber daya

Dokumen terkait:

- Apa itu Traffic Mirroring?
- Menerapkan inspeksi lalu lintas inline dengan menggunakan peralatan keamanan pihak ketiga
- AWS Network Firewall contoh arsitektur dengan routing
- Arsitektur inspeksi terpusat dengan AWS Gateway Load Balancer dan AWS Transit Gateway

Contoh terkait:

- Praktik terbaik untuk men-deploy Penyeimbang Beban Gateway
- TLSkonfigurasi inspeksi untuk lalu lintas jalan keluar terenkripsi dan AWS Network Firewall

Alat terkait:

- AWS Marketplace IDS/IPS

SEC05-BP04 Mengotomatiskan perlindungan jaringan

Otomatiskan penerapan perlindungan jaringan Anda menggunakan DevOps praktik, seperti infrastruktur sebagai kode (IAc) dan pipeline CI/CD. Praktik-praktik ini dapat membantu Anda melacak perubahan dalam perlindungan jaringan Anda melalui sistem kontrol versi, mengurangi waktu yang diperlukan untuk melakukan deployment perubahan, dan membantu Anda untuk mendeteksi apakah perlindungan jaringan Anda menyimpang dari konfigurasi yang Anda inginkan.

Hasil yang diinginkan: Anda menentukan perlindungan jaringan dengan templat dan memasukkannya ke dalam sistem kontrol versi. Pipeline otomatis dimulai ketika perubahan-perubahan baru dibuat yang mengorkestrasi pengujian dan deployment. Pemeriksaan kebijakan dan pengujian statis lainnya diterapkan untuk memvalidasi perubahan sebelum deployment. Anda melakukan deployment perubahan ke lingkungan pentahapan (staging) untuk memvalidasi bahwa berbagai kontrol beroperasi seperti yang diharapkan. Deployment ke lingkungan produksi Anda juga dilakukan secara otomatis setelah kontrol disetujui.

Anti-pola umum:

- Mengandalkan masing-masing tim beban kerja untuk menentukan tumpukan jaringan lengkap, perlindungan, dan otomatisasinya. Tidak memublikasikan aspek-aspek standar dari tumpukan dan perlindungan jaringan secara terpusat yang akan digunakan oleh tim beban kerja.
- Mengandalkan tim jaringan pusat untuk menentukan semua aspek jaringan, perlindungan, dan otomatisasi. Tidak mendelegasikan aspek-aspek spesifik beban kerja dari tumpukan dan perlindungan jaringan kepada tim beban kerja tersebut.
- Mencapai keseimbangan yang tepat antara sentralisasi dan delegasi antara tim jaringan dan tim beban kerja, tetapi tidak menerapkan standar pengujian dan deployment yang konsisten ke seluruh templat IaC dan pipeline CI/CD Anda. Tidak mencatat konfigurasi yang diperlukan dalam peralatan yang memeriksa kepatuhan templat Anda.

Manfaat menerapkan praktik terbaik ini: Menggunakan templat untuk menentukan perlindungan jaringan Anda akan memungkinkan Anda untuk melacak dan membandingkan perubahan dari waktu ke waktu dengan sistem kontrol versi. Penggunaan otomatisasi untuk menguji dan melakukan deployment perubahan akan menghasilkan standardisasi dan prediktabilitas, sehingga akan meningkatkan peluang keberhasilan deployment dan mengurangi konfigurasi manual yang berulang.

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

Panduan implementasi

Sejumlah kontrol perlindungan jaringan yang dijelaskan dalam SEC05-BP02 Mengontrol arus lalu lintas dalam lapisan jaringan Anda dan SEC05-BP03 Menerapkan perlindungan berbasis inspeksi dilengkapi dengan sistem aturan terkelola yang dapat diperbarui secara otomatis berdasarkan intelijen ancaman terbaru. Contoh perlindungan endpoint web Anda termasuk aturan AWS WAF terkelola dan DDoSmitigasi lapisan aplikasi AWS Shield Advanced otomatis. Gunakan grup aturan terkelola AWS Network Firewall untuk tetap up to date dengan daftar domain yang memiliki reputasi rendah dan tanda tangan ancaman juga.

Selain aturan terkelola, kami sarankan Anda menggunakan DevOps praktik untuk mengotomatiskan penerapan sumber daya jaringan, perlindungan, dan aturan yang Anda tentukan. Anda dapat menangkap definisi ini di AWS CloudFormation atau alat infrastruktur sebagai kode (IaC) lain yang Anda pilih, meng-komit-nya ke sistem kontrol versi, dan men-deploynya menggunakan pipeline CI/CD. Gunakan pendekatan ini untuk mendapatkan manfaat tradisional DevOps untuk mengelola kontrol jaringan Anda, seperti rilis yang lebih dapat diprediksi, pengujian otomatis menggunakan alat seperti AWS CloudFormation Guard, dan mendeteksi penyimpangan antara lingkungan yang Anda gunakan dan konfigurasi yang Anda inginkan.

Berdasarkan keputusan yang Anda buat sebagai bagian dari SEC05-BP01 Buat lapisan jaringan, Anda mungkin memiliki pendekatan manajemen pusat untuk membuat VPCs yang didedikasikan untuk arus masuk, keluar, dan inspeksi. Seperti yang dijelaskan dalam Arsitektur Referensi AWS Keamanan (AWS SRA), Anda dapat menentukan ini VPCs dalam akun infrastruktur Jaringan khusus. Anda dapat menggunakan teknik serupa untuk menentukan secara terpusat yang VPCs digunakan oleh beban kerja Anda di akun lain, grup keamanannya, AWS Network Firewall penerapan, aturan Resolver Route 53 dan konfigurasi DNS Firewall, dan sumber daya jaringan lainnya. Anda dapat berbagi sumber daya ini dengan akun Anda yang lain menggunakan AWS Resource Access Manager. Dengan pendekatan ini, Anda dapat menyederhanakan pengujian dan deployment otomatis atas kontrol jaringan Anda ke akun Jaringan, yang bisa Anda lakukan cukup dengan mengelola satu tujuan. Anda dapat melakukannya dalam model hibrida, yang akan memungkinkan Anda melakukan deployment dan membagikan kontrol tertentu secara terpusat serta mendelegasikan kontrol lainnya ke setiap tim beban kerja dan akun mereka masing-masing.

Langkah-langkah implementasi

1. Tetapkan kepemilikan untuk aspek-aspek jaringan dan perlindungan yang ditentukan secara terpusat, dan yang dapat dipelihara oleh tim beban kerja Anda.
2. Buat lingkungan untuk melakukan pengujian dan deployment perubahan-perubahan yang hendak dibuat pada jaringan Anda dan perlindungannya. Misalnya, gunakan akun Pengujian Jaringan dan akun Produksi Jaringan.
3. Tentukan cara Anda akan menyimpan dan memelihara templat Anda dalam sebuah sistem kontrol versi. Simpan templat pusat dalam sebuah repositori yang berbeda dari repositori beban kerja, sedangkan templat beban kerja dapat disimpan dalam repositori-repositori khusus untuk beban kerja tersebut.
4. Buat pipeline CI/CD untuk melakukan pengujian dan deployment templat. Tentukan pengujian untuk memeriksa adanya kesalahan konfigurasi dan apakah templat tersebut mematuhi standar perusahaan Anda, atau tidak.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP06 Mengotomatiskan penerapan kontrol keamanan standar

Dokumen terkait:

- Arsitektur Referensi Keamanan AWS - Akun jaringan

Contoh terkait:

- Arsitektur Referensi Pipeline Deployment AWS
- NetDevSecOpsuntuk memodernisasi penyebaran jaringan AWS
- Mengintegrasikan tes AWS CloudFormation keamanan dengan AWS Security Hub dan laporan AWS CodeBuild

Alat terkait:

- AWS CloudFormation
- AWS CloudFormation Guard
- cfn_nag

Melindungi komputasi

Sumber daya komputasi meliputi instans EC2, kontainer, fungsi AWS Lambda, layanan basis data, perangkat IoT, dan banyak lagi lainnya. Untuk mengamankan tiap-tiap tipe sumber daya komputasi ini, diperlukan pendekatan yang berbeda-beda. Namun demikian, semuanya memiliki strategi yang sama yang perlu Anda pertimbangkan: pertahanan yang mendalam, manajemen kerentanan, pengurangan permukaan serangan, otomatisasi konfigurasi dan operasi, dan melakukan tindakan dari jarak jauh. Pada bagian ini, Anda akan menemukan panduan umum yang bisa dilakukan untuk memproteksi sumber daya komputasi untuk layanan-layanan utama Anda. Untuk tiap-tiap layanan AWS yang digunakan, Anda harus memeriksa saran keamanan spesifik yang diuraikan dalam dokumentasi layanan.

Praktik terbaik

SEC06-BP01 Melakukan manajemen kerentanan

Seringlah memindai dan mem-patch kerentanan pada kode, dependensi, dan infrastruktur Anda untuk membantu mencegah ancaman baru.

Hasil yang diinginkan: Anda memiliki solusi yang terus-menerus memindai beban kerja Anda untuk menemukan kerentanan perangkat lunak, potensi kerusakan, dan paparan jaringan yang tidak diinginkan. Anda telah menetapkan proses dan prosedur untuk mengidentifikasi, memprioritaskan, dan memulihkan kerentanan ini berdasarkan kriteria penilaian risiko. Selain itu, Anda telah menerapkan manajemen patch otomatis untuk instans komputasi Anda. Program manajemen kerentanan Anda terintegrasi ke dalam siklus hidup pengembangan perangkat lunak Anda, dengan solusi untuk memindai kode sumber Anda selama pipeline CI/CD.

Anti-pola 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 pakainya (EOL) dari vendor.
- Melakukan deployment kode ke dalam lingkungan produksi sebelum menganalisis masalah keamanan.

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

Panduan implementasi

Manajemen kerentanan adalah aspek kunci untuk mempertahankan lingkungan cloud yang aman dan andal. Hal ini memerlukan proses komprehensif yang mencakup pemindaian keamanan, identifikasi dan prioritisasi masalah, serta operasi patch untuk mengatasi kerentanan yang

teridentifikasi. Otomatisasi memainkan peran penting dalam proses ini karena memfasilitasi pemindaian beban kerja secara terus-menerus untuk menemukan potensi masalah dan paparan jaringan yang tidak diinginkan, serta upaya remediasi.

Model Tanggung Jawab Bersama AWS adalah konsep mendasar yang mendukung manajemen kerentanan. Menurut model ini, AWS bertanggung jawab untuk mengamankan infrastruktur yang mendasari, termasuk perangkat keras, perangkat lunak, jaringan, dan fasilitas yang menjalankan layanan AWS. Sebaliknya, Anda bertanggung jawab untuk mengamankan data, konfigurasi keamanan, dan tugas manajemen yang terkait dengan layanan, seperti instans Amazon EC2 dan objek Amazon S3.

AWS menawarkan berbagai layanan untuk mendukung program manajemen kerentanan. Amazon Inspector terus memindai beban kerja AWS untuk mencari kerentanan perangkat lunak dan akses jaringan yang tidak diinginkan, sementara Manajer Patch AWS Systems Manager membantu mengelola patching di seluruh instans Amazon EC2. Layanan-layanan ini dapat diintegrasikan dengan AWS Security Hub, layanan manajemen postur keamanan cloud yang mengotomatiskan pemeriksaan keamanan AWS, memusatkan peringatan keamanan, dan memberikan gambaran komprehensif tentang postur keamanan organisasi. Selain itu, Keamanan Amazon CodeGuru menggunakan analisis kode statis untuk mengidentifikasi potensi masalah dalam aplikasi Java dan Python selama fase pengembangan.

Dengan menggabungkan praktik manajemen kerentanan ke dalam siklus hidup pengembangan perangkat lunak, Anda dapat secara proaktif mengatasi kerentanan sebelum diteruskan ke lingkungan produksi, sehingga mengurangi risiko peristiwa keamanan dan meminimalkan potensi dampak kerentanan.

Langkah-langkah implementasi

1. Pahami model tanggung jawab bersama: Tinjau model tanggung jawab bersama AWS untuk memahami tanggung jawab Anda dalam mengamankan beban kerja dan data Anda di cloud. AWS bertanggung jawab untuk mengamankan infrastruktur cloud yang mendasar, sementara Anda bertanggung jawab untuk mengamankan aplikasi, data, dan layanan yang Anda gunakan.
2. Terapkan pemindaian kerentanan: Konfigurasikan layanan pemindaian kerentanan, seperti Amazon Inspector, untuk memindai instans komputasi Anda secara otomatis (misalnya, mesin virtual, kontainer, atau fungsi nirserver) untuk menemukan kerentanan perangkat lunak, potensi kerusakan, dan paparan jaringan yang tidak diinginkan.
3. Tetapkan proses manajemen kerentanan: Tentukan proses dan prosedur untuk mengidentifikasi, memprioritaskan, dan memulihkan kerentanan. Hal ini mungkin mencakup pengaturan jadwal

pemindaian kerentanan reguler, penetapan kriteria penilaian risiko, dan penentuan jadwal remediasi berdasarkan tingkat keparahan kerentanan.

4. Siapkan manajemen patch: Gunakan layanan manajemen patch untuk mengotomatiskan proses patching instans komputasi Anda, baik untuk sistem operasi maupun aplikasi. Anda dapat mengonfigurasi layanan untuk memindai instans guna mengetahui patch yang belum diterapkan dan menginstalnya secara otomatis sesuai jadwal. Pertimbangkan Manajer Patch AWS Systems Manager untuk menyediakan fungsi ini.
5. Konfigurasikan perlindungan malware: Terapkan mekanisme untuk mendeteksi perangkat lunak berbahaya di lingkungan Anda. Misalnya, Anda dapat menggunakan alat seperti Amazon GuardDuty untuk menganalisis, mendeteksi, dan memperingatkan tentang malware dalam volume EC2 dan EBS. GuardDuty juga dapat memindai objek yang baru diunggah ke Amazon S3 untuk mencari potensi malware atau virus dan mengambil tindakan untuk mengisolasinya sebelum terserap ke dalam proses hilir.
6. Integrasikan pemindaian kerentanan dalam pipeline CI/CD: Jika Anda menggunakan pipeline CI/CD untuk deployment aplikasi, integrasikan alat pemindaian kerentanan ke dalam pipeline Anda. Alat seperti Keamanan Amazon CodeGuru dan opsi sumber terbuka dapat memindai kode sumber, dependensi, dan artefak Anda untuk mencari potensi masalah keamanan.
7. Konfigurasikan layanan pemantauan keamanan: Siapkan layanan pemantauan keamanan, seperti AWS Security Hub, untuk mendapatkan gambaran komprehensif tentang postur keamanan Anda di beberapa layanan cloud. Layanan tersebut harus mengumpulkan temuan keamanan dari berbagai sumber dan menyajikannya dalam format standar untuk mempermudah prioritisasi dan remediasi.
8. Terapkan uji penetrasi aplikasi web: Jika aplikasi Anda adalah aplikasi web, dan organisasi Anda memiliki keterampilan yang diperlukan atau dapat menyewa bantuan dari luar, pertimbangkan untuk menerapkan pengujian penetrasi aplikasi web untuk mengidentifikasi potensi kerentanan dalam aplikasi Anda.
9. Otomatiskan dengan infrastruktur sebagai kode: Gunakan alat infrastruktur sebagai kode (IaC), seperti AWS CloudFormation, untuk mengotomatiskan deployment dan konfigurasi sumber daya Anda, termasuk layanan keamanan yang disebutkan sebelumnya. Praktik ini membantu Anda membuat arsitektur sumber daya yang lebih konsisten dan terstandardisasi di beberapa akun dan lingkungan.
10. Pantau dan terus tingkatkan: Terus pantau efektivitas program manajemen kerentanan Anda, dan lakukan peningkatan sesuai kebutuhan. Tinjau temuan keamanan, evaluasi efektivitas upaya remediasi Anda, serta sesuaikan proses dan alat Anda sesuai dengan itu.

Sumber daya

Dokumen terkait:

- AWS Systems Manager
- Gambaran Umum Keamanan AWS Lambda
- Amazon CodeGuru
- Manajemen Kerentanan Otomatis dan Ditingkatkan untuk Beban Kerja Cloud dengan Amazon Inspector Baru
- Mengotomatiskan manajemen kerentanan dan remediasi dalam 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 Komputasi ketentuan dari gambar yang diperkeras

Kurangi peluang untuk akses yang tidak diinginkan ke lingkungan runtime Anda dengan melakukan deployment-nya dari citra yang diperkeras (hardened images). Dapatkan dependensi runtime, seperti citra kontainer dan pustaka aplikasi, hanya dari registri tepercaya dan verifikasikan tanda tangannya. Buat registri privat Anda sendiri untuk menyimpan citra dan pustaka tepercaya yang akan digunakan dalam proses build dan deployment Anda.

Hasil yang diinginkan: Sumber daya komputasi Anda disediakan dari image dasar yang diperkeras. Anda mengambil dependensi eksternal, seperti citra kontainer dan pustaka aplikasi, hanya dari registri tepercaya dan memverifikasi tanda tangannya. Image dan pustaka ini disimpan dalam registri privat untuk dirujuk oleh proses build dan deployment Anda. Anda memindai dan memperbarui citra dan dependensi secara rutin untuk membantu Anda melindungi terhadap kerentanan yang baru ditemukan.

Anti-pola umum:

- Mendapatkan citra dan pustaka dari registri tepercaya, tetapi tidak memverifikasi tanda tangannya atau melakukan pemindaian kerentanan sebelum menggunakannya.

- Melakukan pengerasan citra, tetapi tidak mengujinya secara rutin untuk menemukan kerentanan baru atau memperbaruinya ke versi terkini.
- Menginstal atau tidak menghapus paket-paket perangkat lunak yang tidak diperlukan selama perkiraan siklus hidup citra.
- Hanya mengandalkan patching untuk menjaga sumber daya komputasi produksi tetap mutakhir. Patching saja masih dapat menyebabkan sumber daya komputasi menyimpang dari standar yang diperkeras dari waktu ke waktu. Patching juga dapat gagal menghapus malware yang mungkin telah diinstal oleh pelaku ancaman selama terjadi peristiwa keamanan.

Manfaat menerapkan praktik terbaik ini: Pengerasan gambar akan membantu mengurangi jumlah jalur yang tersedia di lingkungan runtime Anda yang dapat memungkinkan akses yang tidak diinginkan ke pengguna atau layanan yang tidak sah. Hal ini juga dapat mengurangi cakupan dampak jika terjadi akses yang tidak diinginkan.

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

Panduan implementasi

Untuk melakukan pengerasan terhadap sistem Anda, mulailah dari versi sistem operasi, citra kontainer, dan pustaka aplikasi terbaru. Terapkan patch untuk masalah yang diketahui. Minimalkan sistem dengan menghapus aplikasi-aplikasi, layanan, driver perangkat, pengguna default, dan kredensial lainnya yang tidak diperlukan. Lakukan tindakan-tindakan lain yang diperlukan, seperti menonaktifkan port untuk membuat lingkungan yang hanya memiliki sumber daya dan kemampuan yang dibutuhkan oleh beban kerja Anda. Dari acuan dasar ini, Anda kemudian dapat menginstal perangkat lunak, agen, atau proses lain yang Anda butuhkan untuk berbagai tujuan, seperti pemantauan beban kerja atau manajemen kerentanan.

Anda dapat mengurangi beban sistem pengerasan dengan menggunakan panduan yang disediakan sumber tepercaya, seperti Center for Internet Security (CIS) dan Defense Information Systems Agency () Security Technical Implementation Guides (STIGs). DISA Kami menyarankan Anda memulai dengan Amazon Machine Image (AMI) yang diterbitkan oleh AWS atau APN mitra, dan menggunakan AWS EC2Image Builder untuk mengotomatiskan konfigurasi sesuai dengan kombinasi CIS dan STIG kontrol yang sesuai.

Meskipun ada gambar keras yang tersedia dan resep EC2 Image Builder yang menerapkan CIS atau DISA STIG rekomendasi, Anda mungkin menemukan konfigurasi mereka mencegah perangkat lunak Anda berjalan dengan sukses. Dalam situasi ini, Anda dapat memulai dari gambar dasar yang tidak dikeraskan, menginstal perangkat lunak Anda, dan kemudian secara bertahap menerapkan

CIS kontrol untuk menguji dampaknya. Untuk CIS kontrol apa pun yang mencegah perangkat lunak Anda berjalan, uji apakah Anda dapat menerapkan rekomendasi pengerasan berbutir halus sebagai gantinya. DISA Lacak berbagai CIS kontrol dan DISA STIG konfigurasi yang dapat Anda terapkan dengan sukses. Gunakan ini untuk menentukan resep pengerasan gambar Anda di EC2 Image Builder yang sesuai.

Untuk beban kerja kontainer, gambar yang dikeraskan dari Docker tersedia di repositori publik Amazon Elastic Container Registry (). ECR Anda dapat menggunakan EC2 Image Builder untuk mengeraskan gambar kontainer di sampingnyaAMIs.

Mirip dengan sistem operasi dan gambar kontainer, Anda dapat memperoleh paket kode (atau pustaka) dari repositori publik, melalui perkakas seperti pip, npm, Maven, dan. NuGet Kami menyarankan Anda untuk mengelola paket kode dengan mengintegrasikan repositori pribadi, seperti dalam AWS CodeArtifact, dengan repositori publik tepercaya. Integrasi ini dapat menangani pengambilan, penyimpanan, dan penyimpanan paket up-to-date untuk Anda. Proses pembuatan aplikasi Anda kemudian dapat memperoleh dan menguji versi terbaru dari paket-paket ini bersama aplikasi Anda, menggunakan teknik seperti Analisis Komposisi Perangkat Lunak (SCA), Pengujian Keamanan Aplikasi Statis (SAST), dan Pengujian Keamanan Aplikasi Dinamis (DAST).

Untuk beban kerja tanpa server yang digunakan AWS Lambda, sederhanakan pengelolaan dependensi paket menggunakan lapisan Lambda. Gunakan lapisan Lambda untuk mengonfigurasi sekumpulan dependensi standar yang ada di berbagai fungsi ke dalam arsip mandiri. Anda dapat membuat dan memelihara lapisan melalui proses pembuatannya sendiri, menyediakan cara sentral agar fungsi Anda tetap ada up-to-date.

Langkah-langkah implementasi

- Lakukan pengerasan terhadap sistem operasi. Gunakan gambar dasar dari sumber tepercaya sebagai fondasi untuk membangun pengerasan Anda. AMIs Gunakan EC2Image Builder untuk membantu menyesuaikan perangkat lunak yang diinstal pada gambar Anda.
- Lakukan pengerasan terhadap sumber daya terkontainerisasi. Konfigurasikan sumber daya terkontainerisasi untuk memenuhi praktik-praktik terbaik keamanan. Saat menggunakan kontainer, terapkan Pemindaian ECR Gambar di pipeline build Anda dan secara teratur terhadap repositori gambar Anda untuk dicari CVEs di container Anda.
- Saat menggunakan implementasi tanpa server dengan AWS Lambda, gunakan lapisan Lambda untuk memisahkan kode fungsi aplikasi dan pustaka dependen bersama. Konfigurasikan penandatanganan kode untuk Lambda untuk memastikan bahwa hanya kode tepercaya yang berjalan dalam fungsi Lambda Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP05 Lakukan manajemen tambalan

Video terkait:

- Menyelam jauh ke dalam AWS Lambda keamanan

Contoh terkait:

- Membangun STIG -compliant dengan cepat AMI menggunakan EC2 Image Builder
- Membangun image kontainer yang lebih baik
- Menggunakan lapisan Lambda untuk menyederhanakan proses pengembangan Anda
- Mengembangkan & Menyebarkan AWS Lambda Layers menggunakan Serverless Framework
- Membangun pipa end-to-end AWS DevSecOps CI/CD dengan open sourceSCA, SAST dan alat DAST

SEC06-BP03 Mengurangi manajemen manual dan akses interaktif

Gunakan otomatisasi sebisa mungkin untuk melakukan tugas deployment, konfigurasi, pemeliharaan, dan investigasi. Pertimbangkan akses manual ke sumber daya komputasi saat melakukan prosedur darurat atau berada dalam lingkungan yang aman (sandbox), ketika otomatisasi tidak tersedia.

Hasil yang diinginkan: Skrip terprogram dan dokumen otomatisasi (runbook) merekam tindakan resmi pada sumber daya komputasi Anda. Runbook ini dimulai baik secara otomatis, melalui sistem deteksi perubahan, maupun secara manual, ketika penilaian oleh manusia diperlukan. Akses langsung ke sumber daya komputasi hanya tersedia dalam situasi darurat ketika otomatisasi tidak tersedia. Semua aktivitas manual dicatat lognya dan dimasukkan ke dalam proses peninjauan untuk terus meningkatkan kemampuan-kemampuan otomatisasi Anda.

Anti-pola umum:

- Akses interaktif ke EC2 instans Amazon dengan protokol seperti atau. SSH RDP
- Mempertahankan login pengguna individu seperti `/etc/passwd` atau pengguna lokal Windows.
- Berbagi kata sandi atau kunci privat untuk mengakses instans di antara beberapa pengguna.

- Menginstal perangkat lunak dan membuat atau memperbarui file konfigurasi secara manual.
- Memperbarui atau melakukan patching perangkat lunak secara manual.
- Masuk ke instans untuk memecahkan masalah.

Manfaat menerapkan praktik terbaik ini: Melakukan tindakan dengan otomatisasi dapat membantu Anda mengurangi risiko operasional dari perubahan dan kesalahan konfigurasi yang tidak diinginkan. Menghapus penggunaan Secure Shell (SSH) dan Remote Desktop Protocol (RDP) untuk akses interaktif mengurangi cakupan akses ke sumber daya komputasi Anda. Hal ini akan menghilangkan jalur umum yang memungkinkan tindakan tidak sah. Pencatatan tugas manajemen sumber daya komputasi Anda dalam dokumen otomatisasi dan skrip programatis akan menyediakan sebuah mekanisme untuk menentukan dan mengaudit cakupan penuh aktivitas yang sah secara lebih mendetail.

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

Panduan implementasi

Masuk ke instans adalah pendekatan klasik untuk administrasi sistem. Setelah menginstal sistem operasi server, pengguna biasanya akan masuk log in secara manual untuk mengonfigurasi sistem dan menginstal perangkat lunak yang diinginkan. Selama masa pakai server, pengguna mungkin masuk log in untuk melakukan pembaruan perangkat lunak, menerapkan patch, mengubah konfigurasi, dan memecahkan masalah.

Namun demikian, akses secara manual dapat menimbulkan sejumlah risiko. Ini membutuhkan server yang mendengarkan permintaan, seperti SSH atau RDP layanan, yang dapat memberikan jalur potensial untuk akses yang tidak sah. Ini juga meningkatkan risiko kesalahan manusia yang terkait dengan pelaksanaan langkah-langkah secara manual. Hal ini dapat mengakibatkan insiden beban kerja, kerusakan atau pemusnahan data, atau masalah-masalah keamanan lainnya. Akses manusia juga memerlukan perlindungan dari tindakan berbagi kredensial, yang bisa menimbulkan biaya overhead manajemen tambahan.

Untuk mengurangi risiko ini, Anda dapat menerapkan solusi akses jarak jauh berbasis agen, seperti Systems Manager.AWS AWS Systems Manager Agen (SSMAgen) memulai saluran terenkripsi dan dengan demikian tidak bergantung pada mendengarkan permintaan yang dimulai secara eksternal. Pertimbangkan untuk mengonfigurasi SSM Agen untuk membuat saluran ini melalui titik VPC akhir.

Systems Manager memberikan Anda kontrol terperinci tentang cara berinteraksi dengan instans terkelola. Anda dapat menentukan otomatisasi yang akan dijalankan, siapa yang bisa

menjalankannya, dan kapan mereka bisa menjalankannya. Systems Manager dapat menerapkan patch, menginstal perangkat lunak, dan membuat perubahan konfigurasi tanpa akses interaktif ke instans. Systems Manager juga dapat menyediakan akses ke shell jarak jauh dan mencatat setiap perintah yang dipanggil, dan outputnya, selama sesi ke log dan Amazon S3. AWS CloudTrailmencatat pemanggilan Systems Manager APIs untuk inspeksi.

Langkah-langkah implementasi

1. Instal Agen AWS Systems Manager (SSMAgen) di EC2 instans Amazon Anda. Periksa untuk melihat apakah SSM Agen disertakan dan dimulai secara otomatis sebagai bagian dari AMI konfigurasi dasar Anda.
2. Verifikasi bahwa IAM Peran yang terkait dengan profil EC2 instans Anda menyertakan IAMkebijakan `AmazonSSMManagedInstanceCore` terkelola.
3. Nonaktifkan SSHRDP,, dan layanan akses jarak jauh lainnya yang berjalan pada instans Anda. Anda dapat melakukan ini dengan menjalankan skrip yang dikonfigurasi di bagian data pengguna dari templat peluncuran Anda atau dengan membangun yang disesuaikan AMIs dengan alat seperti EC2 Image Builder.
4. Verifikasi bahwa aturan masuknya grup keamanan yang berlaku untuk EC2 instans Anda tidak mengizinkan akses pada port 22/tcp (SSH) atau port 3389/tcp (). RDP Implementasikan deteksi dan peringatan terkait grup keamanan yang salah konfigurasi dengan menggunakan layanan-layanan seperti AWS Config.
5. Tentukan otomatisasi dan runbook yang sesuai, lalu jalankan perintah di Systems Manager. Gunakan IAM kebijakan untuk menentukan siapa yang dapat melakukan tindakan ini dan kondisi di mana mereka diizinkan. Uji otomatisasi ini secara menyeluruh di sebuah lingkungan non-produksi. Lakukan invokasi terhadap otomatisasi ini jika diperlukan, bukan mengakses instans secara interaktif.
6. Gunakan AWS Systems Manager Session Manager untuk menyediakan akses interaktif ke instans bila diperlukan. Aktifkan pencatatan aktivitas sesi untuk mempertahankan jejak audit di Amazon CloudWatch Log atau Amazon S3.

Sumber daya

Praktik-praktik terbaik terkait:

- REL08-BP04 Terapkan menggunakan infrastruktur yang tidak dapat diubah

Contoh terkait:

- Mengganti SSH akses untuk mengurangi overhead manajemen dan keamanan dengan AWS Systems Manager

Alat terkait:

- AWS Systems Manager

Video terkait:

- Mengontrol Akses Sesi Pengguna ke Instans di Manajer AWS Systems Manager Sesi

SEC06-BP04 Validasi integritas perangkat lunak

Gunakan verifikasi kriptografis untuk memvalidasi integritas artefak perangkat lunak (termasuk citra) yang digunakan beban kerja Anda. Tanda tangani perangkat lunak Anda secara kriptografis sebagai perlindungan terhadap perubahan-perubahan tidak sah yang berjalan di lingkungan komputasi Anda.

Hasil yang diinginkan: Semua artefak diperoleh dari sumber-sumber yang tepercaya. Sertifikat situs web vendor divalidasi. Artefak yang diunduh sudah diverifikasi secara kriptografis berdasarkan tanda tangannya. Perangkat lunak Anda sendiri ditandatangani secara kriptografis dan diverifikasi oleh lingkungan-lingkungan komputasi Anda.

Anti-pola umum:

- Memercayai situs web vendor terkemuka untuk mendapatkan artefak perangkat lunak, tetapi mengabaikan pemberitahuan tentang sertifikat yang kedaluwarsa. Melanjutkan pengunduhan tanpa mengonfirmasi bahwa sertifikatnya valid.
- Memvalidasi sertifikat situs web vendor, tetapi tidak secara kriptografis memverifikasi artefak yang diunduh dari situs web ini.
- Hanya mengandalkan digest atau hash untuk memvalidasi integritas perangkat lunak. Hash menetapkan bahwa artefak belum dimodifikasi dari versi aslinya, tetapi tidak memvalidasi sumbernya.
- Tidak menandatangani perangkat lunak, kode, atau pustaka Anda sendiri, meskipun hanya digunakan dalam deployment Anda sendiri.

Manfaat menerapkan praktik terbaik ini: Memvalidasi integritas artefak yang bergantung pada beban kerja Anda akan membantu Anda dalam mencegah malware memasuki lingkungan komputasi Anda. Menandatangani perangkat lunak Anda akan membantu melindungi dari eksekusi yang tidak sah di lingkungan komputasi Anda. Amankan rantai pasokan perangkat lunak Anda dengan menandatangani dan memverifikasi kodenya.

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

Panduan implementasi

Citra sistem operasi, citra kontainer, dan artefak kode sering kali didistribusikan dengan pemeriksaan integritas yang tersedia, seperti melalui digest atau hash. Hal ini memungkinkan klien untuk memverifikasi integritas dengan melakukan komputasi hash atas payload mereka sendiri dan memastikan itu sama dengan yang dipublikasikan. Meskipun pemeriksaan ini membantu Anda untuk memverifikasi bahwa muatan belum dirusak, namun pemeriksaan tersebut tidak memvalidasi muatan yang berasal dari sumber aslinya (asalnya). Verifikasi asal-usul mengharuskan adanya sertifikat yang dikeluarkan oleh otoritas tepercaya untuk menandatangani artefaknya secara digital.

Jika Anda menggunakan sebuah perangkat lunak atau artefak unduhan dalam beban kerja Anda, periksa apakah penyedianya menyediakan kunci publik untuk verifikasi tanda tangan digital. Berikut ini adalah beberapa contoh cara AWS menyediakan kunci publik dan instruksi verifikasi untuk perangkat lunak yang kami publikasikan:

- EC2Image Builder: Verifikasi tanda tangan unduhan AWS TOE instalasi
- AWS Systems Manager: Memverifikasi tanda tangan SSM Agen
- Amazon CloudWatch: Memverifikasi tanda tangan paket CloudWatch agen

Masukkan verifikasi tanda tangan digital ke dalam proses yang Anda gunakan untuk memperoleh dan mengeraskan gambar, seperti yang dibahas dalam SEC06-BP02 Komputasi ketentuan dari gambar yang diperkeras.

Anda dapat menggunakannya AWS Signer untuk membantu Anda mengelola verifikasi tanda tangan, serta siklus hidup penandatanganan kode Anda sendiri untuk perangkat lunak dan artefak Anda sendiri. Keduanya, AWS Lambda dan Amazon Elastic Container Registry menyediakan integrasi dengan Signer untuk memverifikasi tanda tangan kode dan image Anda. Menggunakan contoh di bagian Sumber Daya, Anda dapat menerapkan Signer ke dalam pipeline integrasi dan pengiriman berkelanjutan (CI/CD) untuk mengotomatiskan verifikasi tanda tangan serta penandatanganan kode dan image Anda sendiri.

Sumber daya

Dokumen terkait:

- Penandatanganan Kriptografi untuk Kontainer
- Praktik Terbaik untuk membantu mengamankan pipeline build image container Anda dengan menggunakan AWS Signer
- Mengumumkan Penandatanganan Gambar Kontainer dengan AWS Signer dan Amazon EKS
- Mengkonfigurasi penandatanganan kode untuk AWS Lambda
- Praktik-praktik terbaik dan pola lanjutan untuk penandatanganan kode Lambda
- Penandatanganan kode menggunakan CA AWS Certificate Manager Pribadi dan kunci AWS Key Management Service asimetris

Contoh terkait:

- Otomatiskan penandatanganan kode Lambda dengan Amazon dan CodeCatalyst AWS Signer
- Menandatangani dan Memvalidasi OCI Artefak dengan AWS Signer

Alat terkait:

- AWS Lambda
- AWS Signer
- AWS Certificate Manager
- AWS Key Management Service
- AWS CodeArtifact

SEC06-BP05 Mengotomatiskan perlindungan komputasi

Lakukan otomatisasi terhadap operasi perlindungan komputasi untuk mengurangi kebutuhan akan intervensi manusia. Gunakan pemindaian otomatis untuk mendeteksi adanya potensi masalah yang mungkin terjadi dalam sumber daya komputasi Anda, dan lakukan remediasi dengan respons programatis otomatis atau operasi manajemen armada. Gabungkan otomatisasi dalam proses CI/CD Anda untuk menyebarkan beban kerja yang dapat dipercaya dengan dependensi. up-to-date

Hasil yang diinginkan: Sistem otomatis melakukan semua pemindaian dan penambalan atas sumber daya komputasi. Anda menggunakan verifikasi otomatis untuk memeriksa apakah gambar dan dependensi perangkat lunak berasal dari sumber tepercaya, dan belum dirusak. Beban kerja secara otomatis diperiksa untuk up-to-date dependensi, dan ditandatangani untuk membangun kepercayaan di lingkungan komputasi. AWS Remediasi otomatis dimulai ketika ada sumber daya yang tidak mematuhi persyaratan terdeteksi.

Anti-pola umum:

- Mengikuti praktik infrastruktur tak dapat diubah, tetapi tidak memiliki solusi untuk melakukan patching atau penggantian darurat sistem produksi.
- Menggunakan otomatisasi untuk memperbaiki sumber daya yang salah konfigurasi, tetapi tidak memiliki mekanisme untuk melakukan penimpaan secara manual. Kesulitan mungkin akan muncul saat Anda perlu menyesuaikan persyaratan, dan Anda mungkin perlu menangguhkan otomatisasi sampai Anda membuat perubahan-perubahan ini.

Manfaat menerapkan praktik terbaik ini: Otomatisasi dapat mengurangi risiko akses dan penggunaan sumber daya komputasi Anda yang tidak sah. Hal ini akan membantu Anda mencegah terjadinya kesalahan konfigurasi yang masuk ke lingkungan produksi, serta mendeteksi dan memperbaiki kesalahan konfigurasi jika terjadi. Otomatisasi juga akan membantu Anda mendeteksi akses dan penggunaan yang tidak sah atas sumber daya komputasi untuk mempercepat waktu respons Anda. Hal ini pada gilirannya dapat mengurangi cakupan dampak yang ditimbulkan masalah secara keseluruhan.

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

Panduan implementasi

Anda dapat menerapkan otomatisasi yang dijelaskan dalam praktik Pilar Keamanan untuk memberikan proteksi terhadap melindungi sumber daya komputasi Anda. SEC06-BP01 Melakukan manajemen kerentanan menjelaskan cara Anda dapat menggunakan Amazon Inspector di pipeline CI/CD Anda dan untuk terus memindai lingkungan runtime Anda untuk Kerentanan Umum dan Eksposur () yang diketahui. CVEs Anda dapat menggunakan AWS Systems Manager untuk menerapkan pacth atau melakukan deployment ulang atas image gambar baru melalui runbook otomatis agar armada komputasi Anda diperbarui dengan perangkat lunak dan pustaka terbaru. Gunakan teknik-teknik ini untuk mengurangi kebutuhan akan proses-proses manual dan akses interaktif ke sumber daya komputasi Anda. Lihat SEC06-BP03 Mengurangi manajemen manual dan akses interaktif untuk mempelajari lebih lanjut.

Otomasi juga berperan dalam menyebarkan beban kerja yang dapat dipercaya, dijelaskan dalam SEC06-BP02 Komputasi ketentuan dari gambar yang diperkeras dan 06-BP04 Validasi integritas perangkat lunak. SEC Anda dapat menggunakan layanan seperti EC2Image Builder,, AWS SignerAWS CodeArtifact, dan Amazon Elastic Container Registry (ECR) untuk mengunduh, memverifikasi, membuat, dan menyimpan gambar dan dependensi kode yang diperkeras dan disetujui. Di samping Inspector, masing-masing dapat berperan dalam proses CI/CD Anda sehingga beban kerja Anda masuk ke produksi hanya ketika dikonfirmasi bahwa dependensinya berasal dan dari sumber tepercaya. up-to-date Beban kerja Anda juga ditandatangani sehingga lingkungan AWS komputasi, seperti AWS Lambdadan Amazon Elastic EKS Kubernetes Service () dapat memverifikasi bahwa itu belum dirusak sebelum mengizinkannya dijalankan.

Selain kontrol-kontrol preventif ini, Anda juga dapat menggunakan otomatisasi dalam kontrol-kontrol detektif untuk sumber daya komputasi Anda. Sebagai salah satu contoh, AWS Security Hubmenawarkan standar NIST800-53 Rev. 5 yang mencakup pemeriksaan seperti EC2instance [EC2.8] harus menggunakan Instance Metadata Service Version 2 (). IMDSv2 IMDSv2menggunakan teknik otentikasi sesi, memblokir permintaan yang berisi X-Forwarded-For HTTP header, dan jaringan TTL 1 untuk menghentikan lalu lintas yang berasal dari sumber eksternal untuk mengambil informasi tentang instance. EC2 Pemeriksaan di Security Hub ini dapat mendeteksi kapan EC2 instans menggunakan IMDSv1 dan memulai remediasi otomatis. Pelajari lebih lanjut tentang deteksi dan remediasi otomatis di SEC04-BP04 Memulai remediasi untuk sumber daya yang tidak sesuai.

Langkah-langkah implementasi

1. Otomatiskan pembuatan yang aman, sesuai, dan diperkeras dengan Image AMIs BuilderEC2. Anda dapat menghasilkan gambar yang menggabungkan kontrol dari standar Center for Internet Security (CIS) Benchmark atau Security Technical Implementation Guide (STIG) dari gambar dasar AWS dan APN mitra.
2. Melakukan otomatisasi manajemen konfigurasi. Berlakukan dan validasikan konfigurasi-konfigurasi yang aman di sumber daya komputasi Anda secara otomatis dengan menggunakan layanan atau alat manajemen konfigurasi.
   a. Manajemen konfigurasi otomatis menggunakan AWS Config
   b. Keamanan otomatis dan manajemen postur kepatuhan dengan menggunakan AWS Security Hub
3. Otomatiskan penambalan atau penggantian instans Amazon Elastic Compute Cloud (AmazonEC2). AWS Systems Manager Patch Manager mengotomatiskan proses menambal instance terkelola dengan pembaruan terkait keamanan dan jenis pembaruan lainnya. Anda dapat menggunakan Patch Manager untuk menerapkan patch untuk kedua sistem operasi dan aplikasi.

   a. AWS Manajer Patch Systems Manager
4. Otomatiskan pemindaian sumber daya komputasi untuk kerentanan dan eksposur umum (CVEs), dan sematkan solusi pemindaian keamanan dalam pipeline build Anda.
   a. Amazon Inspector
   b. ECRPemindaian Gambar
5. Pertimbangkan Amazon GuardDuty untuk malware otomatis dan deteksi ancaman untuk melindungi sumber daya komputasi. GuardDuty juga dapat mengidentifikasi potensi masalah ketika suatu AWS Lambdafungsi dipanggil di AWS lingkungan Anda.
   a. Amazon GuardDuty
6. Pertimbangkan solusi AWS Mitra. AWS Mitra menawarkan produk terdepan di industri yang setara, identik, atau terintegrasi dengan kontrol yang ada di lingkungan lokal Anda. Produk-produk ini akan melengkapi layanan-layanan AWS yang ada untuk memungkinkan Anda melakukan deployment arsitektur keamanan yang menyeluruh dan pengalaman yang lebih lancar di seluruh lingkungan cloud dan on-premise Anda.
   a. Keamanan infrastruktur

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP06 Mengotomatiskan penerapan kontrol keamanan standar

Dokumen terkait:

- Dapatkan manfaat penuh IMDSv2 dan nonaktifkan IMDSv1 di seluruh AWS infrastruktur Anda

Video terkait:

- Praktik terbaik keamanan untuk layanan EC2 metadata instans Amazon

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 Pahami skema klasifikasi data Anda

Pahami klasifikasi data yang diproses beban kerja Anda, persyaratan penanganannya, proses-proses bisnis terkait, di mana data disimpan, dan siapa pemilik data. Skema klasifikasi dan penanganan data Anda harus mempertimbangkan persyaratan-persyaratan hukum dan kepatuhan yang berlaku atas beban kerja Anda serta kontrol data apa yang diperlukan. Pemahaman terhadap data adalah langkah pertama dalam perjalanan klasifikasi data.

Hasil yang diinginkan: Jenis data yang ada dalam beban kerja Anda dapat dipahami dan didokumentasikan dengan baik. Kontrol yang tepat diterapkan untuk melindungi data sensitif berdasarkan klasifikasinya. Kontrol ini mengatur berbagai pertimbangan, seperti siapa yang diizinkan mengakses data dan untuk tujuan apa, di mana data disimpan, kebijakan enkripsi yang diterapkan untuk data tersebut dan bagaimana kunci enkripsi dikelola, siklus hidup untuk data dan persyaratan retensinya, proses pemusnahan yang tepat, proses pencadangan dan pemulihan apa yang diterapkan, serta audit akses.

Anti-pola umum:

- Tidak memiliki kebijakan klasifikasi data formal untuk menentukan tingkat sensitivitas data dan persyaratan-persyaratan penanganannya
- Tidak memiliki pemahaman yang baik tentang tingkat sensitivitas data dalam beban kerja Anda, dan tidak merekam informasi ini dalam dokumentasi arsitektur dan operasi
- Gagal menerapkan kontrol-kontrol yang sesuai terhadap data Anda berdasarkan sensitivitas dan persyaratannya, sebagaimana yang diuraikan dalam kebijakan klasifikasi dan penanganan data Anda
- Gagal memberikan umpan balik tentang persyaratan-persyaratan klasifikasi dan penanganan data kepada pemilik kebijakan.

Manfaat menjalankan praktik terbaik ini: Praktik ini akan menghilangkan ambiguitas yang mungkin muncul di seputar penanganan data yang tepat dalam beban kerja Anda. Penerapan kebijakan formal yang menentukan tingkat sensitivitas data di organisasi Anda dan perlindungan yang diperlukan dapat membantu Anda dalam mematuhi peraturan hukum serta pengesahan dan sertifikasi keamanan siber lainnya. Pemilik beban kerja dapat merasa yakin dengan mengetahui di mana data sensitif disimpan dan kontrol perlindungan apa yang diterapkan. Dengan merekam hal ini dalam dokumentasi, anggota tim baru akan dapat lebih memahaminya dan dapat memelihara berbagai kontrol di awal masa kerja mereka. Praktik-praktik ini juga dapat membantu Anda mengurangi biaya dengan melakukan penyesuaian ukuran yang tepat terhadap kontrol untuk masing-masing jenis data.

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

Panduan implementasi

Saat merancang sebuah beban kerja, Anda mungkin mempertimbangkan cara untuk melindungi data sensitif secara intuitif. Misalnya, dalam sebuah aplikasi multi-tenant, data setiap tenant secara intuitif

dianggap sebagai data sensitif dan perlindungan diterapkan agar satu tenant tidak dapat mengakses data tenant yang lain. Demikian juga, Anda dapat secara intuitif merancang kontrol akses sehingga hanya administrator yang dapat melakukan modifikasi data, sedangkan pengguna yang lain hanya memiliki akses tingkat baca atau tidak memiliki akses sama sekali.

Dengan menetapkan dan merekam tingkat sensitivitas data ini dalam kebijakan, bersama dengan persyaratan-persyaratan perlindungan datanya, Anda dapat secara formal mengidentifikasi data apa yang berada dalam beban kerja Anda. Anda kemudian dapat menentukan apakah kontrol yang tepat sudah diterapkan, apakah kontrol dapat diaudit, dan respons apa yang sesuai jika ternyata data salah ditangani.

Untuk membantu mengidentifikasi di mana data sensitif berada dalam beban kerja Anda, pertimbangkan untuk menggunakan katalog data. Katalog data adalah basis data yang memetakan data di organisasi Anda, lokasinya, tingkat sensitivitas, dan kontrol yang ada untuk melindungi data tersebut. Selain itu, pertimbangkan untuk menggunakan tanda sumber daya jika tersedia. Misalnya, Anda dapat menerapkan tanda yang memiliki kunci tanda `Classification` dan nilai tanda `PHI` untuk informasi kesehatan yang dilindungi (PHI), dan tanda lain yang memiliki kunci tanda `Sensitivity` dan nilai tanda `High`. Layanan-layanan seperti AWS Config, kemudian dapat digunakan untuk memantau sumber daya ini untuk mendeteksi adanya perubahan dan peringatan jika sumber daya tersebut dimodifikasi dengan cara yang membuatnya melanggar kepatuhan terhadap persyaratan perlindungan Anda (seperti mengubah pengaturan enkripsi). Anda dapat merekam definisi standar kunci tanda dan nilai yang dapat diterima menggunakan kebijakan tanda, fitur dari AWS Organizations. Sebaiknya kunci atau nilai tanda tidak berisi data privat atau sensitif.

Langkah-langkah implementasi

1. Pahami skema klasifikasi dan persyaratan-persyaratan perlindungan data organisasi Anda.
2. Identifikasi jenis data sensitif yang diproses oleh beban kerja Anda.
3. Rekam data dalam katalog data yang menyediakan gambaran tunggal tentang di mana data berada dalam organisasi dan tingkat sensitivitas data tersebut.
4. Pertimbangkan untuk menggunakan penandaan tingkat sumber daya dan data, jika tersedia, untuk memberikan tanda pada data dengan tingkat sensitivitasnya dan metadata operasional lainnya yang dapat membantu Anda memantau dan merespons insiden.
   a. Kebijakan tanda AWS Organizations dapat digunakan untuk memberlakukan standar penandaan.

Sumber daya

Praktik-praktik terbaik terkait:

- SUS04-BP01 Mengimplementasikan kebijakan klasifikasi data

Dokumen terkait:

- Laporan Resmi Klasifikasi Data
- Praktik Terbaik untuk Penandaan Sumber Daya AWS

Contoh terkait:

- Sintaks dan Contoh Kebijakan Tanda AWS Organizations

Alat terkait

- Editor Tanda AWS

SEC07-BP02 Terapkan kontrol perlindungan data berdasarkan sensitivitas data

Terapkan kontrol-kontrol perlindungan data yang memberikan tingkat kontrol yang sesuai untuk setiap kelas data yang ditentukan dalam kebijakan klasifikasi Anda. Praktik ini dapat memungkinkan Anda untuk melindungi data sensitif dari akses dan penggunaan yang tidak sah, sekaligus menjaga ketersediaan dan penggunaan data.

Hasil yang diinginkan: Anda memiliki kebijakan klasifikasi yang mendefinisikan berbagai tingkat sensitivitas untuk data yang ada dalam organisasi Anda. Untuk masing-masing tingkat sensitivitas ini, Anda memiliki pedoman yang jelas yang dipublikasikan untuk layanan dan lokasi penyimpanan dan penanganan yang disetujui, serta konfigurasi yang diperlukan. Anda mengimplementasikan kontrol untuk masing-masing tingkat sesuai dengan tingkat perlindungan yang diperlukan dan biaya yang terkait. Anda memiliki pemantauan dan peringatan untuk mendeteksi apakah data ada di lokasi yang tidak sah, diproses di lingkungan yang tidak sah, diakses oleh pelaku yang tidak sah, atau konfigurasi layanan terkait tidak lagi mematuhi persyaratan yang ditetapkan.

Anti-pola umum:

- Menerapkan tingkat kontrol perlindungan yang sama terhadap semua data. Hal ini dapat menyebabkan penyediaan kontrol keamanan yang berlebihan untuk data yang tidak sensitif, atau perlindungan yang tidak memadai untuk data yang sangat sensitif.
- Tidak melibatkan para pemangku kepentingan yang relevan dari tim keamanan, kepatuhan, dan bisnis saat menentukan kontrol perlindungan data.
- Mengabaikan biaya overhead operasional dan biaya yang terkait dengan implementasi dan pemeliharaan kontrol perlindungan data.
- Tidak melakukan peninjauan kontrol perlindungan data secara berkala untuk menjaga keselarasan dengan kebijakan klasifikasi.
- Tidak memiliki inventaris lengkap terkait di mana data berada saat diam dan saat bergerak.

Manfaat menjalankan praktik terbaik ini: Dengan menyelaraskan kontrol Anda dengan tingkat klasifikasi data Anda, organisasi Anda akan dapat berinvestasi dalam tingkat kontrol yang lebih tinggi jika diperlukan. Hal ini dapat mencakup penambahan sumber daya untuk pengamanan, pemantauan, pengukuran, remediasi, dan pelaporan. Jika kontrol yang diperlukan lebih sedikit, maka Anda dapat meningkatkan aksesibilitas dan kelengkapan data untuk tenaga kerja, pelanggan, atau konstituen Anda. Pendekatan ini akan memberikan organisasi Anda fleksibilitas penggunaan data yang paling besar, sekaligus tetap mematuhi persyaratan-persyaratan perlindungan data.

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

Panduan implementasi

Mengimplementasikan kontrol-kontrol perlindungan data berdasarkan tingkat sensitivitas data mencakup beberapa langkah penting. Pertama, mengidentifikasi tingkat sensitivitas data yang berbeda-beda dalam arsitektur beban kerja Anda (seperti publik, internal, rahasia, dan terbatas) dan kemudian mengevaluasi di mana Anda menyimpan dan memproses data ini. Selanjutnya, tentukan batasan-batasan isolasi di sekitar data berdasarkan tingkat sensitivitasnya. Kami menyarankan Anda untuk memisahkan data ke dalam Akun AWS yang berbeda, dengan menggunakan kebijakan kontrol layanan (SCP) untuk membatasi layanan dan tindakan yang diizinkan untuk setiap tingkat sensitivitas data. Dengan cara ini, Anda dapat membuat batasan-batasan isolasi yang kuat dan memberlakukan prinsip hak akses paling rendah.

Setelah Anda menentukan batasan-batasan isolasi tersebut, implementasikan kontrol-kontrol perlindungan yang sesuai berdasarkan tingkat sensitivitas data. Lihat praktik terbaik untuk Melindungi data diam dan Melindungi data bergerak untuk menerapkan kontrol yang relevan seperti enkripsi, kontrol akses, dan audit. Pertimbangkan teknik-teknik seperti tokenisasi atau anonimisasi untuk

mengurangi tingkat sensitivitas data Anda. Sederhanakan penerapan kebijakan data yang konsisten di seluruh bisnis Anda dengan sistem tersentralisasi untuk tokenisasi dan detokenisasi.

Lakukan pemantauan dan pengujian secara terus-menerus terhadap efektivitas kontrol yang diimplementasikan. Lakukan peninjauan dan pembaruan secara rutin terhadap skema klasifikasi data, penilaian risiko, dan kontrol perlindungan karena lanskap data dan ancaman bagi organisasi Anda terus berubah. Selaraskan kontrol-kontrol perlindungan data yang diimplementasikan dengan peraturan industri, standar, dan persyaratan hukum yang relevan. Selanjutnya, berikan pengetahuan dan pelatihan keamanan untuk membantu para karyawan memahami skema klasifikasi data serta tanggung jawab mereka dalam menangani dan melindungi data sensitif.

Langkah-langkah implementasi

1. Identifikasi tingkat klasifikasi dan sensitivitas data yang ada dalam beban kerja Anda.
2. Tentukan batasan-batasan isolasi untuk setiap tingkat dan tentukan strategi penegakannya.
3. Lakukan evaluasi terhadap kontrol-kontrol yang Anda tetapkan yang mengatur akses, enkripsi, audit, retensi, dan lainnya yang diwajibkan berdasarkan kebijakan klasifikasi data Anda.
4. Lakukan evaluasi terhadap opsi-opsi untuk mengurangi tingkat sensitivitas data jika sesuai, seperti menggunakan tokenisasi atau anonimisasi.
5. Pastikan bahwa kontrol Anda menggunakan pengujian dan pemantauan otomatis terhadap sumber daya yang dikonfigurasi.

Sumber daya

Praktik-praktik terbaik terkait:

- PERF03-BP01 Menggunakan penyimpanan data yang dibuat khusus yang paling mendukung persyaratan akses data dan penyimpanan data Anda
- COST04-BP05 Menegakkan kebijakan retensi data

Dokumen terkait:

- Laporan Resmi Klasifikasi Data
- Praktik Terbaik untuk Keamanan, Identitas & Kepatuhan
- Praktik Terbaik AWS KMS
- Praktik dan fitur terbaik enkripsi untuk layanan AWS

Contoh terkait:

- Membangun solusi tokenisasi nirserver untuk melakukan masking terhadap data sensitif
- Cara menggunakan tokenisasi untuk meningkatkan keamanan data dan mengurangi cakupan audit

Alat terkait:

- AWS Key Management Service (AWS KMS)
- AWS CloudHSM
- AWS Organizations

SEC07-BP03 Otomatiskan identifikasi dan klasifikasi

Otomatisasi identifikasi dan klasifikasi data dapat membantu Anda mengimplementasikan kontrol yang tepat. Penggunaan otomatisasi untuk melengkapi proses penentuan manual akan mengurangi risiko terjadinya kesalahan manusia dan paparan.

Hasil yang diinginkan: Anda dapat melakukan verifikasi apakah kontrol yang tepat sudah dilakukan berdasarkan klasifikasi dan kebijakan penanganan Anda. Alat-alat dan layanan otomatis dapat membantu Anda mengidentifikasi dan mengklasifikasikan tingkat sensitivitas data Anda. Otomatisasi juga akan membantu Anda untuk terus memantau lingkungan Anda guna mendeteksi dan memperingatkan jika data sedang disimpan atau sedang ditangani secara tidak sah sehingga Anda bisa melakukan tindakan korektif dengan cepat.

Anti-pola umum:

- Hanya mengandalkan proses-proses manual untuk melakukan identifikasi dan klasifikasi data, yang bisa jadi rawan kesalahan dan memakan waktu. Hal ini dapat menyebabkan klasifikasi data yang tidak efisien dan tidak konsisten, terutama saat volume data semakin besar.
- Tidak memiliki mekanisme untuk melacak dan mengelola aset data yang ada di seluruh organisasi.
- Mengabaikan perlunya pemantauan dan klasifikasi data yang berkelanjutan seiring pergerakan dan perkembangan data di dalam organisasi.

Manfaat menjalankan praktik terbaik ini: Melakukan otomatisasi atas identifikasi dan klasifikasi data dapat mengantarkan Anda pada penerapan kontrol perlindungan data yang lebih konsisten dan akurat, mengurangi risiko terjadinya kesalahan manusia. Otomatisasi juga dapat memberikan

visibilitas terhadap akses dan pergerakan data sensitif, sehingga akan membantu Anda untuk mendeteksi penanganan yang tidak sah dan mengambil tindakan korektif.

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

Panduan implementasi

Meskipun penilaian manusia sering kali digunakan untuk mengklasifikasikan data selama fase desain awal beban kerja, Anda harus mempertimbangkan untuk memiliki sistem yang mengotomatiskan identifikasi dan klasifikasi terhadap data uji sebagai sebuah kontrol preventif. Misalnya, developer dapat diberi sebuah alat atau layanan untuk memindai data representatif guna menentukan sensitivitasnya. Di AWS, Anda dapat mengunggah kumpulan data ke Amazon S3 dan memindainya menggunakan Amazon Macie, Amazon Comprehend, atau Amazon Comprehend Medical. Selain itu, pertimbangkan juga untuk melakukan pemindaian data sebagai bagian dari pengujian unit dan integrasi untuk mendeteksi di mana data sensitif tidak diharapkan. Mengirim peringatan terkait data sensitif pada tahap ini dapat menyoroti adanya kesenjangan dalam perlindungan sebelum dilakukan deployment ke produksi. Fitur lain seperti deteksi data sensitif di AWS Glue, Amazon SNS, dan Amazon CloudWatch juga dapat digunakan untuk mendeteksi PII dan mengambil tindakan mitigasi. Untuk alat atau layanan otomatis apa pun, pahami cara alat atau layanan tersebut menentukan data sensitif, kemudian lengkapi dengan solusi manusia atau solusi otomatis lainnya untuk mengatasi kesenjangan apa pun sesuai kebutuhan.

Sebagai sebuah kontrol pendeteksi, gunakan pemantauan berkelanjutan terhadap lingkungan Anda untuk mendeteksi apakah data sensitif saat ini disimpan dengan cara yang tidak mematuhi persyaratan, atau tidak. Hal ini dapat membantu Anda mendeteksi berbagai kesulitan, seperti data sensitif yang dikirimkan ke file log atau disalin ke lingkungan analitik data tanpa penghapusan atau penyamaran identitas yang tepat. Data yang disimpan di Amazon S3 dapat terus dipantau untuk menemukan data sensitif menggunakan Amazon Macie.

Langkah-langkah implementasi

1. Tinjau skema klasifikasi data di organisasi Anda yang dijelaskan dalam SEC07-BP01.
    a. Dengan pemahaman terhadap skema klasifikasi data organisasi Anda, Anda dapat menetapkan proses yang akurat untuk identifikasi dan klasifikasi otomatis yang selaras dengan kebijakan perusahaan Anda.
2. Lakukan pemindaian awal terhadap lingkungan Anda untuk identifikasi dan klasifikasi otomatis.
    a. Pemindaian penuh di awal terhadap data Anda dapat membantu menghasilkan pemahaman komprehensif tentang di mana data sensitif berada di lingkungan Anda. Jika pemindaian penuh

pada awalnya tidak diperlukan atau tidak dapat diselesaikan di awal karena biaya, evaluasi apakah teknik-teknik pengambilan sampel data sudah cocok untuk meraih hasil-hasil yang Anda tetapkan. Misalnya, Amazon Macie dapat dikonfigurasi untuk melakukan operasi penemuan data sensitif otomatis secara meluas di seluruh bucket S3 Anda. Kemampuan ini menggunakan teknik-teknik pengambilan sampel untuk melakukan analisis awal terkait di mana data sensitif berada dengan cara yang hemat. Analisis bucket S3 yang lebih mendalam kemudian dapat dilakukan dengan menggunakan pekerjaan penemuan data sensitif. Penyimpanan data lainnya juga dapat diekspor ke S3 untuk dipindai oleh Macie.

   b. Tetapkan kontrol akses yang ditentukan dalam SEC07-BP02 untuk sumber daya penyimpanan data Anda yang diidentifikasi dalam pemindaian Anda.

3. Konfigurasikan pemindaian yang berkelanjutan terhadap lingkungan Anda.
   a. Kemampuan penemuan data sensitif otomatis yang dimiliki Macie dapat digunakan untuk melakukan pemindaian yang berkelanjutan terhadap lingkungan Anda. Bucket S3 yang diketahui yang diotorisasi untuk menyimpan data sensitif dapat dikecualikan menggunakan daftar yang diizinkan di Macie.
4. Terapkan identifikasi dan klasifikasi ke dalam proses build dan pengujian Anda.
   a. Identifikasi alat-alat yang dapat digunakan developer untuk memindai data guna menentukan sensitivitasnya saat beban kerja sedang dikembangkan. Gunakan alat-alat ini sebagai bagian dari pengujian integrasi untuk memberikan peringatan ketika ada data sensitif yang tidak terduga dan mencegah deployment lebih lanjut.
5. Implementasikan sebuah sistem atau runbook untuk melakukan tindakan ketika data sensitif ditemukan di lokasi yang tidak sah.
   a. Batasi akses ke data menggunakan remediasi otomatis. Misalnya, Anda dapat memindahkan data ini ke bucket S3 dengan akses terbatas atau menandai objek jika Anda menggunakan kontrol akses berbasis atribut (ABAC). Selain itu, pertimbangkan untuk melakukan masking data saat terdeteksi.
   b. Peringatkan tim perlindungan data dan respons insiden Anda untuk menyelidiki akar penyebab insiden tersebut. Pembelajaran apa pun yang mereka identifikasi dapat membantu mencegah insiden di masa depan.

Sumber daya

Dokumen terkait:

- AWS Glue: Mendeteksi dan memproses data sensitif

- Menggunakan pengidentifikasi data terkelola di Amazon SNS
- Log Amazon CloudWatch: Membantu melindungi data log sensitif dengan melakukan masking

Contoh terkait:

- Mengaktifkan klasifikasi data untuk basis data Amazon RDS dengan Macie
- Mendeteksi data sensitif di DynamoDB menggunakan Macie

Alat terkait:

- Amazon Macie
- Amazon Comprehend
- Amazon Comprehend Medical
- AWS Glue

SEC07-BP04 Tentukan manajemen siklus hidup data yang dapat diskalakan

Pahami persyaratan-persyaratan siklus hidup data Anda karena persyaratan tersebut terkait dengan berbagai tingkat klasifikasi dan penanganan data Anda. Hal ini dapat mencakup cara data ditangani ketika pertama kali memasuki lingkungan Anda, cara data ditransformasi, dan aturan untuk pemusnahannya. Pertimbangkan faktor-faktor seperti periode retensi, akses, audit, dan pelacakan asal.

Hasil yang diinginkan: Anda mengklasifikasikan data sedekat mungkin dengan titik dan waktu konsumsi. Ketika klasifikasi data memerlukan proses masking, tokenisasi, atau proses-proses lain yang mengurangi tingkat sensitivitas, Anda harus melakukan tindakan-tindakan ini sedekat mungkin dengan titik dan waktu penyerapan.

Anda menghapus data sesuai dengan kebijakan Anda ketika data tersebut tidak lagi layak untuk dipertahankan, berdasarkan klasifikasinya.

Anti-pola umum:

- Mengimplementasikan satu pendekatan umum terhadap manajemen siklus hidup data, tanpa mempertimbangkan berbagai tingkat sensitivitas dan persyaratan akses yang berbeda-beda.

- Mempertimbangkan manajemen siklus hidup hanya dari perspektif data yang dapat digunakan, atau data yang dicadangkan, tetapi tidak keduanya.
- Menganggap data yang telah memasuki beban kerja Anda sebagai data yang valid, tanpa mengetahui nilai atau asal-usulnya.
- Mengandalkan daya tahan data sebagai pengganti untuk pencadangan dan perlindungan data.
- Mempertahankan data melampaui masa kegunaannya dan periode retensi yang diperlukan.

Manfaat menjalankan praktik terbaik ini: Strategi manajemen siklus hidup data yang ditentukan dengan baik dan dapat diskalakan akan membantu Anda dalam menjaga kepatuhan terhadap peraturan, meningkatkan keamanan data, mengoptimalkan biaya penyimpanan, dan memungkinkan akses dan berbagi data yang efisien sekaligus mempertahankan kontrol yang tepat.

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

Panduan implementasi

Data dalam beban kerja sering kali bersifat dinamis. Bentuk data saat memasuki lingkungan beban kerja Anda dapat berbeda-beda, dari ketika data disimpan atau digunakan dalam logika bisnis, pelaporan, analitik, atau machine learning. Selain itu, nilai data dapat berubah seiring waktu. Beberapa data bersifat temporal dan kehilangan nilai seiring umurnya bertambah. Pertimbangkan dampak dari berbagai perubahan data Anda ini terhadap evaluasi berdasarkan skema klasifikasi data Anda dan kontrol-kontrol terkait. Jika memungkinkan, gunakan mekanisme siklus hidup otomatis, seperti kebijakan siklus hidup Amazon S3 dan Amazon Data Lifecycle Manager, untuk mengonfigurasi proses retensi data, pengarsipan, dan kedaluwarsa data Anda. Untuk data yang disimpan di DynamoDB, Anda dapat menggunakan fitur Time To Live (TTL) untuk menentukan stempel waktu kedaluwarsa per item.

Bedakan antara data yang tersedia untuk digunakan, dan data yang disimpan sebagai cadangan. Pertimbangkan untuk menggunakan AWS Backup untuk mengotomatiskan cadangan data di seluruh layanan AWS. Snapshot Amazon EBS menyediakan cara untuk menyalin volume EBS dan menyimpannya menggunakan fitur S3, termasuk siklus hidup, perlindungan data, dan akses ke mekanisme perlindungan. Dua dari mekanisme ini adalah Kunci Objek S3 dan AWS Backup Kunci Vault, yang dapat memberi Anda keamanan dan kontrol tambahan atas cadangan Anda. Kelola pemisahan tugas dan akses yang jelas untuk cadangan. Isolasi cadangan di tingkat akun agar tetap terpisah dari lingkungan yang terpengaruh saat ada suatu peristiwa yang terjadi.

Aspek lain dari manajemen siklus hidup adalah merekam riwayat data saat berlangsung melalui beban kerja Anda, yang disebut pelacakan asal data. Pelacakan ini dapat memberikan keyakinan

bahwa Anda tahu dari mana data berasal, setiap transformasi yang dilakukan, pemilik atau proses apa yang membuat perubahan tersebut, dan kapan. Riwayat ini dapat membantu Anda dalam melakukan pemecahan masalah dan investigasi selama peristiwa keamanan yang mungkin terjadi. Misalnya, Anda dapat mencatat log metadata tentang transformasi dalam sebuah tabel Amazon DynamoDB. Dalam sebuah danau data, Anda dapat menyimpan salinan data yang ditransformasi di dalam bucket S3 yang berbeda untuk setiap tahap pipeline data. Simpan informasi skema dan stempel waktu dalam file AWS Glue Data Catalog. Terlepas dari solusi yang Anda gunakan, pertimbangkan kebutuhan pengguna akhir Anda untuk menentukan peralatan yang tepat yang Anda butuhkan untuk melaporkan asal-usul data Anda. Hal ini akan membantu Anda menentukan cara terbaik dalam melacak asal-usul data Anda.

Langkah-langkah implementasi

1. Analisis jenis data, tingkat sensitivitas, dan persyaratan akses beban kerja untuk mengklasifikasikan data dan menentukan strategi-strategi manajemen siklus hidup yang sesuai.
2. Rancang dan implementasikan kebijakan retensi data dan proses pemusnahan otomatis yang selaras dengan persyaratan-persyaratan berdasarkan hukum, peraturan, dan organisasi.
3. Tetapkan proses dan otomatisasi untuk melakukan pemantauan, audit, dan penyesuaian berkelanjutan terhadap strategi, kontrol, dan kebijakan manajemen siklus hidup data seiring dengan perubahan persyaratan beban kerja dan perubahan peraturan.
   a. Deteksi sumber daya yang tidak memiliki manajemen siklus hidup otomatis yang aktif dengan AWS Config

Sumber daya

Praktik-praktik terbaik terkait:

- COST04-BP05 Menegakkan kebijakan retensi data
- SUS04-BP03 Menggunakan kebijakan untuk mengelola siklus hidup set data Anda

Dokumen terkait:

- Laporan Resmi Klasifikasi Data
- Cetak Biru AWS untuk Pertahanan Ransomware
- Panduan DevOps: Meningkatkan ketertelusuran dengan pelacakan asal data

Contoh terkait:

- Bagaimana melindungi data sensitif untuk seluruh siklus hidupnya di AWS
- Membangun garis keturunan data untuk danau data menggunakan AWS Glue, Amazon Neptune, dan Spline

Alat terkait:

- AWS Backup
- Amazon Data Lifecycle Manager
- AWS Identity and Access Management Access Analyzer

Lindungi data diam

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.

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.

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 Log 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 Otomatiskan perlindungan data diam
- SEC08-BP04 Menerapkan kontrol akses

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: Anda memiliki mekanisme manajemen kunci yang dapat diskalakan, dapat diulang, dan dapat diotomatiskan. Mekanisme ini menerapkan hak akses paling rendah ke materi kunci dan memberikan keseimbangan yang tepat antara ketersediaan, kerahasiaan, dan integritas kunci. Anda memantau akses ke kunci, dan jika rotasi materi kunci diperlukan, Anda merotasinya menggunakan proses otomatis. Anda tidak mengizinkan materi kunci diakses oleh operator manusia.

Anti-pola umum:

- Akses manusia ke materi kunci yang tidak dienkripsi.
- 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-persyaratan berdasarkan peraturan untuk mengenkripsi data Anda. Solusi manajemen kunci yang efektif

dapat memberikan mekanisme-mekanisme teknis yang selaras dengan peraturan tersebut untuk melindungi materi kunci.

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

Panduan implementasi

Enkripsi data diam adalah kontrol keamanan mendasar. Untuk menerapkan kontrol ini, beban kerja Anda memerlukan sebuah mekanisme untuk secara aman 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 diintegrasikan 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 melakukan deployment beban kerja menggunakan strategi multi-akun, Anda harus menyimpan kunci AWS KMS di akun yang sama dengan beban kerja yang menggunakannya. Model terdistribusi ini memberikan tanggung jawab untuk mengelola kunci AWS KMS kepada tim Anda. Dalam kasus penggunaan lainnya, organisasi Anda dapat memilih untuk menyimpan kunci AWS KMS ke dalam sebuah akun tersentralisasi. Struktur tersentralisasi ini memerlukan kebijakan tambahan untuk mengaktifkan akses lintas akun yang diperlukan agar akun beban kerja dapat mengakses kunci yang disimpan di akun tersentralisasi 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, Anda harus mengontrol akses ke kunci dengan ketat melalui penggunaan kebijakan kunci dan kebijakan IAM. Kebijakan kunci adalah cara utama untuk mengontrol akses ke sebuah kunci AWS KMS. Selain itu, pemberian kunci AWS KMS dapat memberikan akses ke layanan AWS untuk mengenkripsi dan mendekripsi data atas nama Anda. Tinjau panduan untuk kontrol akses ke kunci AWS KMS Anda.

Anda harus memantau penggunaan kunci enkripsi untuk mendeteksi pola-pola akses yang tidak biasa. Operasi-operasi yang dijalankan menggunakan kunci yang dikelola AWS dan kunci yang dikelola pelanggan yang disimpan AWS KMS dapat dicatat log-nya di AWS CloudTrail dan harus ditinjau secara berkala. Berikan perhatian khusus untuk memantau peristiwa pemusnahan kunci. Untuk memitigasi pemusnahan materi kunci yang tidak disengaja atau tidak sah, peristiwa pemusnahan kunci tidak akan langsung menghapus materi kunci tersebut. Percobaan untuk menghapus kunci yang ada di AWS KMS akan mengikuti periode tunggu, yang secara default diatur

ke 30 hari dan minimal 7 hari, sehingga memberikan waktu kepada administrator untuk meninjau tindakan ini dan membatalkan permintaannya 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 yang dikelola pelanggan. Jika beban kerja Anda memerlukan penggunaan langsung AWS KMS untuk mengenkripsi atau mendekripsi data, Anda harus menggunakan enkripsi amplop untuk melindungi data Anda. AWS Encryption SDK dapat menyediakan primitif enkripsi di sisi klien untuk aplikasi Anda guna mengimplementasikan enkripsi amplop dan mengintegrasikannya dengan AWS KMS.

Langkah-langkah implementasi

1. Tentukan opsi manajemen kunci yang sesuai (dikelola AWS atau dikelola pelanggan) untuk kunci tersebut.
   a. Untuk memudahkan penggunaan, AWS menawarkan kunci yang dimiliki AWS dan kunci yang dikelola AWS untuk sebagian besar layanan, yang menyediakan kemampuan enkripsi data diam tanpa perlu mengelola materi kunci atau kebijakan kunci.
   b. Saat menggunakan kunci yang dikelola oleh 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 tersebut. 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.
   a. Cara layanan AWS menggunakan AWS KMS
   b. Untuk informasi mendetail tentang berbagai opsi enkripsi yang ditawarkan oleh layanan AWS, lihat topik Enkripsi Diam dalam panduan pengguna atau panduan developer untuk layanan tersebut.
3. Implementasikan AWS KMS: AWS KMS memudahkan Anda untuk membuat dan mengelola kunci serta mengontrol penggunaan enkripsi di berbagai layanan AWS dan di dalam aplikasi Anda.
   a. Memulai: AWS Key Management Service (AWS KMS)
   b. Tinjau praktik-praktik terbaik untuk kontrol akses ke kunci AWS KMS Anda.
4. Pertimbangkan AWS Encryption SDK: Gunakan AWS Encryption SDK dengan integrasi AWS KMS jika aplikasi Anda harus mengenkripsi data di sisi klien.

   a. AWS Encryption SDK

5. Aktifkan IAM Access Analyzer agar secara otomatis meninjau dan memberi tahu jika ada kebijakan kunci AWS KMS yang terlalu luas.
   a. Pertimbangkan untuk menggunakan pemeriksaan kebijakan kustom untuk memverifikasi bahwa pembaruan kebijakan sumber daya tidak memberikan akses publik ke Kunci KMS.
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.
   a. Beberapa organisasi lebih suka melakukan segregasi terhadap aktivitas pencatatan log AWS KMS ke dalam jejak terpisah. Untuk membaca detail selengkapnya, lihat bagian Mencatat Log Panggilan API AWS KMS dengan CloudTrail dari panduan developer AWS KMS.

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

Enkripsikan data pribadi saat diam untuk menjaga kerahasiaan dan memberikan lapisan perlindungan tambahan terhadap pengungkapan atau eksfiltrasi data yang tidak diinginkan. Enkripsi melindungi data sehingga tidak dapat dibaca atau diakses tanpa didekripsi terlebih dahulu. Inventarisasi dan kontrol data yang tidak terenkripsi untuk mengurangi risiko yang terkait dengan paparan data.

Hasil yang diinginkan: Anda memiliki mekanisme yang mengenkripsi data pribadi secara default saat diam. mekanisme ini membantu menjaga kerahasiaan data dan memberikan lapisan perlindungan tambahan terhadap pengungkapan atau eksfiltrasi data yang tidak disengaja. Anda memelihara inventaris data yang tidak terenkripsi dan memahami kontrol yang ada untuk melindunginya.

Anti-pola umum:

- Tidak menggunakan konfigurasi yang dienkripsi 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 diterapkan: Tinggi

Panduan implementasi

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

AWS Key Management Service (AWS KMS) terintegrasi dengan berbagai layanan-layanan AWS untuk mempermudah enkripsi data diam. Misalnya, di Amazon Elastic Compute Cloud (Amazon EC2), Anda dapat mengatur enkripsi default pada akun sehingga volume EBS 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 menggunakan kunci yang dikelola pelanggan (CMK). Anda memiliki kontrol penuh atas CMK, termasuk rotasi dan manajemen akses melalui penggunaan kebijakan-kebijakan kunci.

Selain itu, layanan seperti Amazon Simple Storage Service (Amazon S3) sekarang mengenkripsi semua objek baru secara default. Implementasi ini memberikan keamanan yang ditingkatkan tanpa berdampak pada kinerja.

Layanan lain, seperti Amazon Elastic Compute Cloud (Amazon EC2) atau Amazon Elastic File System (Amazon EFS), mendukung pengaturan untuk enkripsi default. Anda juga dapat menggunakan Aturan AWS Config untuk memeriksa secara otomatis apakah Anda menggunakan enkripsi untuk volume Amazon Elastic Block Store (Amazon EBS), instans Amazon Relational Database Service (Amazon RDS), dan bucket Amazon S3, serta layanan lainnya dalam organisasi Anda.

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

Langkah-langkah implementasi

- Konfigurasikan enkripsi default untuk volume Amazon EBS baru: Tentukan bahwa Anda ingin agar semua volume EBS baru dibuat 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 yang dikonfigurasi akan mengenkripsi volume root dan snapshot secara otomatis.

- Konfigurasikan enkripsi Amazon RDS: Konfigurasikan enkripsi untuk klaster dan snapshot basis data Anda saat diam dengan menggunakan opsi enkripsi.
- Buat dan konfigurasikan kunci AWS KMS dengan kebijakan yang membatasi akses ke principal yang sesuai untuk masing-masing klasifikasi data: Misalnya, buat satu kunci AWS KMS untuk mengenkripsi data produksi dan kunci lain untuk mengenkripsi data pengembangan atau pengujian. Anda juga dapat menyediakan kunci akses ke Akun AWS lainnya. Pertimbangkan untuk memiliki akun yang berbeda untuk lingkungan-lingkungan pengembangan dan produksi Anda. Jika lingkungan produksi Anda perlu mendekripsi artefak yang ada di akun pengembangan, Anda dapat mengedit kebijakan CMK yang digunakan untuk mengenkripsi artefak pengembangan agar akun produksi dapat mendekripsi artefak tersebut. Dan kemudian lingkungan produksi dapat menyerap data yang didekripsi untuk digunakan dalam lingkungan produksi.
- Konfigurasikan enkripsi di layanan AWS tambahan: Untuk layanan-layanan AWS lain yang Anda gunakan, tinjau dokumentasi keamanan untuk layanan-layanan tersebut guna menentukan opsi enkripsi layanan.

Sumber daya

Dokumen terkait:

- Alat Kripto 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 Otomatiskan perlindungan data diam

Gunakan otomatisasi untuk memvalidasi dan menerapkan kontrol-kontrol data diam. Gunakan pemindaian otomatis untuk mendeteksi kesalahan konfigurasi pada solusi-solusi penyimpanan data Anda, dan lakukan remediasi melalui respons programatis otomatis jika memungkinkan. Terapkan otomatisasi dalam proses CI/CD Anda untuk mendeteksi kesalahan konfigurasi penyimpanan data sebelum di-deploy ke lingkungan produksi.

Hasil yang diinginkan: Sistem otomatis memindai dan memantau lokasi penyimpanan data untuk mencari adanya kesalahan konfigurasi kontrol, akses tidak sah, dan penggunaan yang tidak terduga. Deteksi lokasi penyimpanan yang salah konfigurasi akan memulai remediasi otomatis. Proses-proses otomatis membuat cadangan data dan menyimpan salinan yang tak bisa diubah di luar lingkungan asli.

Anti-pola umum:

- Tidak mempertimbangkan opsi untuk mengaktifkan enkripsi berdasarkan pengaturan-pengaturan default, jika didukung.
- Tidak mempertimbangkan peristiwa keamanan, selain peristiwa operasional, saat merumuskan strategi pencadangan dan pemulihan otomatis.
- Tidak menerapkan pengaturan akses publik untuk layanan-layanan penyimpanan.
- Tidak memantau dan mengaudit kontrol Anda untuk melindungi data diam.

Manfaat menjalankan praktik terbaik ini: Otomatisasi akan membantu Anda dalam mencegah risiko terjadinya kesalahan konfigurasi lokasi penyimpanan data Anda. Hal ini membantu Anda untuk mencegah kesalahan konfigurasi memasuki lingkungan produksi Anda. Praktik terbaik ini juga membantu Anda untuk mendeteksi dan memperbaiki kesalahan konfigurasi, jika terjadi.

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

Panduan implementasi

Otomatisasi adalah tema yang ada di seluruh praktik untuk melindungi data diam Anda. SEC01-BP06 Mengotomatiskan deployment kontrol keamanan standar menjelaskan bagaimana Anda dapat merekam konfigurasi sumber daya Anda dengan menggunakan templat infrastruktur sebagai kode (IaC), seperti dengan AWS CloudFormation. Templat ini di-commit ke sistem kontrol versi, dan digunakan untuk melakukan deployment sumber daya di AWS melalui sebuah pipeline CI/CD.

Teknik-teknik ini juga berlaku untuk mengotomatiskan konfigurasi solusi penyimpanan data Anda, seperti pengaturan enkripsi di bucket Amazon S3.

Anda dapat memeriksa pengaturan yang Anda tentukan di templat IaC untuk menemukan kesalahan konfigurasi pada pipeline CI/CD Anda menggunakan aturan di AWS CloudFormation Guard. Anda dapat memantau pengaturan yang belum tersedia di CloudFormation atau perkakas IaC lainnya untuk mencari kesalahan konfigurasi dengan AWS Config. Peringatan yang dihasilkan Config untuk kesalahan konfigurasi dapat diperbaiki secara otomatis, seperti yang dijelaskan dalam SEC04-BP04 Mulai melakukan remediasi untuk sumber daya yang tidak mematuhi persyaratan.

Penggunaan otomatisasi sebagai bagian dari strategi manajemen izin Anda juga merupakan komponen integral dari perlindungan data otomatis. SEC03-BP02 Memberikan hak akses paling rendah dan SEC03-BP04 Mengurangi izin secara terus menerus menjelaskan konfigurasi kebijakan akses hak akses paling rendah yang secara terus dipantau oleh AWS Identity and Access Management Access Analyzer untuk menghasilkan temuan ketika izin dapat dikurangi. Selain otomatisasi untuk izin pemantauan, Anda dapat mengonfigurasi Amazon GuardDuty untuk mengawasi perilaku akses data anomali untuk volume EBS Anda (melalui instans EC2), bucket S3, dan basis data Amazon Relational Database Service yang didukung.

Otomatisasi juga berperan dalam mendeteksi ketika ada data sensitif yang disimpan di lokasi yang tidak sah. SEC07-BP03 Identifikasi dan klasifikasi otomatis menjelaskan cara Amazon Macie dapat memantau bucket S3 Anda untuk mencari data sensitif yang tidak terduga dan menghasilkan peringatan yang dapat memulai respons otomatis.

Ikuti praktik-praktik yang diuraikan di REL09 Cadangkan data untuk mengembangkan strategi pencadangan dan pemulihan data otomatis. Pencadangan dan pemulihan data sama pentingnya untuk pemulihan dari peristiwa-peristiwa keamanan seperti halnya untuk peristiwa operasional.

Langkah-langkah implementasi

1. Tetapkan konfigurasi penyimpanan data dalam templat IaC. Gunakan pemeriksaan otomatis di pipeline CI/CD Anda untuk mendeteksi terjadinya kesalahan konfigurasi.
   a. Untuk AWS CloudFormation, Anda dapat menggunakan templat IaC, dan AWS CloudFormation Guard guna memeriksa kesalahan konfigurasi pada templat.
   b. Gunakan AWS Config untuk menjalankan aturan-aturan dalam mode evaluasi proaktif. Gunakan pengaturan ini untuk memeriksa kepatuhan sumber daya sebagai sebuah langkah dalam pipeline CI/CD Anda sebelum membuatnya.
2. Pantau sumber daya untuk menemukan kesalahan konfigurasi penyimpanan data.

   a. Setel AWS Config untuk memantau sumber daya penyimpanan data untuk perubahan dalam konfigurasi kontrol dan menghasilkan peringatan untuk menginvokasi tindakan remediasi saat mendeteksi kesalahan konfigurasi.
   b. Lihat SEC04-BP04 Mulai melakukan remediasi untuk sumber daya yang tidak mematuhi persyaratan untuk panduan lebih lanjut tentang cara melakukan remediasi otomatis.
3. Pantau dan kurangi izin akses data secara berkelanjutan melalui otomatisasi.
   a. IAM Access Analyzer dapat berjalan secara terus menerus untuk menghasilkan peringatan ketika izin berpotensi dikurangi.
4. Pantau dan beri peringatan tentang terjadinya perilaku akses data yang tidak normal.
   a. GuardDuty akan mengamati tanda tangan ancaman yang diketahui dan penyimpangan dari perilaku akses dasar untuk sumber daya penyimpanan data seperti volume EBS, bucket S3, dan basis data RDS.
5. Pantau dan beri peringatan tentang adanya data sensitif yang disimpan di lokasi yang tidak diharapkan.
   a. Gunakan Amazon Macie untuk memindai bucket S3 Anda secara terus menerus untuk mencari data sensitif.
6. Otomatiskan pencadangan yang aman dan terenkripsi terhadap data Anda.
   a. AWS Backup adalah sebuah layanan terkelola yang memungkinkan Anda membuat cadangan dari berbagai sumber data di AWS. Elastic Disaster Recovery memungkinkan Anda untuk menyalin beban kerja server penuh dan mempertahankan perlindungan data berkelanjutan dengan sasaran titik pemulihan (RPO) yang diukur dalam hitungan detik. Anda dapat mengonfigurasi kedua layanan tersebut untuk bekerja bersama dalam mengotomatiskan pembuatan cadangan data dan menyalinnya ke lokasi-lokasi failover. Hal ini dapat membantu menjaga data Anda tetap tersedia saat terkena dampak peristiwa operasional atau keamanan.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP06 Otomatiskan deployment kontrol keamanan standar
- SEC03-BP02 Memberikan hak akses paling rendah
- SEC03-BP04 Mengurangi izin secara terus-menerus
- SEC04-BP04 Mulai melakukan remediasi untuk sumber daya yang tidak mematuhi persyaratan
- SEC07-BP03 Otomatiskan identifikasi dan klasifikasi

- REL09-BP02 Mengamankan dan mengenkripsikan cadangan
- REL09-BP03 Melakukan pencadangan data secara otomatis

Dokumen terkait:

- Panduan Preskriptif AWS: Secara otomatis mengenkripsi volume Amazon EBS yang ada dan yang baru
- Manajemen Risiko Ransomware di AWS dengan Menggunakan Kerangka Keamanan Siber (CSF) NIST

Contoh terkait:

- Cara menggunakan aturan proaktif AWS Config dan AWS CloudFormation Hooks untuk mencegah pembuatan sumber daya cloud yang tidak sesuai
- Mengotomatiskan dan mengelola perlindungan data secara terpusat untuk Amazon S3 dengan menggunakan AWS Backup
- AWS re:Invent 2023 - Menerapkan perlindungan data proaktif menggunakan snapshot Amazon EBS
- AWS re:Invent 2022 - Membangun dan mengotomatiskan ketahanan dengan perlindungan data modern

Alat terkait:

- AWS CloudFormation Guard
- Registri Aturan AWS CloudFormation Guard
- IAM Access Analyzer
- Amazon Macie
- AWS Backup
- Elastic Disaster Recovery

SEC08-BP04 Menerapkan kontrol akses

Untuk membantu melindungi data diam, terapkan kontrol akses menggunakan mekanisme, seperti isolasi dan penentuan versi. Terapkan hak akses paling rendah dan kontrol akses bersyarat. Cegah pemberian akses publik ke data Anda.

Hasil yang diinginkan: Anda memverifikasi bahwa hanya pengguna yang berwenang saja yang dapat mengakses data berdasarkan kebutuhan untuk mengetahui. Anda melindungi data Anda dengan pencadangan dan penentuan versi rutin untuk mencegah pengubahan atau penghapusan data yang disengaja atau tidak disengaja. Anda mengisolasi data penting dari data lain untuk melindungi kerahasiaan dan integritas data tersebut.

Anti-pola 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 diterapkan: Tinggi

Panduan implementasi

Melindungi data diam penting untuk menjaga integritas, kerahasiaan, dan kepatuhan data terhadap persyaratan peraturan. Anda dapat menerapkan beberapa kontrol untuk membantu mencapai hal ini, termasuk kontrol akses, isolasi, akses bersyarat, dan penentuan versi.

Anda dapat memberlakukan kontrol akses dengan prinsip hak akses paling rendah, sehingga hanya memberikan izin yang diperlukan kepada pengguna dan layanan untuk melakukan tugasnya. Ini termasuk akses ke kunci enkripsi. Tinjau kebijakan AWS Key Management Service (AWS KMS) Anda untuk memverifikasi bahwa tingkat akses yang Anda berikan sudah sesuai dan ketentuan yang relevan berlaku.

Anda dapat memisahkan data berdasarkan berbagai tingkat klasifikasi dengan menggunakan Akun AWS khusus untuk setiap tingkat, dan mengelola akun ini menggunakan AWS Organizations. Isolasi ini dapat membantu mencegah akses yang tidak sah dan meminimalkan risiko paparan data.

Tinjau tingkat akses yang diberikan dalam kebijakan bucket S3 secara rutin. Hindari menggunakan bucket yang dapat dibaca atau ditulis secara publik kecuali jika benar-benar diperlukan. Pertimbangkan untuk menggunakan AWS Config untuk mendeteksi bucket yang tersedia secara publik dan Amazon CloudFront untuk menyajikan konten dari Amazon S3. Pastikan bucket yang seharusnya tidak mengizinkan akses publik telah dikonfigurasi dengan benar untuk mencegahnya.

Terapkan mekanisme penentuan versi dan penguncian objek untuk data penting yang disimpan di Amazon S3. Penentuan versi Amazon S3 mempertahankan versi objek sebelumnya untuk memulihkan data dari penghapusan atau penimpaan yang tidak disengaja. Kunci Objek Amazon S3 menyediakan kontrol akses wajib untuk objek, sehingga mencegahnya dihapus atau ditimpa, bahkan oleh pengguna root, hingga kunci ini kedaluwarsa. Selain itu, Kunci Vault Amazon S3 Glacier menawarkan fitur serupa untuk arsip yang disimpan di Amazon S3 Glacier.

Langkah-langkah implementasi

1. Berlakukan kontrol akses dengan prinsip hak akses paling rendah:
    - Tinjau izin akses yang diberikan kepada pengguna dan layanan, dan verifikasi bahwa izin tersebut memang diperlukan untuk melakukan tugasnya.
    - Tinjau akses ke kunci enkripsi dengan memeriksa kebijakan AWS Key Management Service (AWS KMS).
2. Pisahkan data berdasarkan berbagai tingkat klasifikasi:
    - Gunakan Akun AWS khusus untuk setiap tingkat klasifikasi data.
    - Kelola akun ini menggunakan AWS Organizations.
3. Tinjau izin bucket dan objek Amazon S3:
    - Tinjau tingkat akses yang diberikan dalam kebijakan bucket S3 secara rutin.
    - Hindari menggunakan bucket yang dapat dibaca atau ditulis secara publik kecuali jika benar-benar diperlukan.
    - Pertimbangkan untuk menggunakan AWS Config guna mendeteksi bucket yang tersedia untuk umum.
    - Gunakan Amazon CloudFront untuk menyajikan konten dari Amazon S3.
    - Pastikan bucket yang seharusnya tidak mengizinkan akses publik telah dikonfigurasi dengan benar untuk mencegahnya.
    - Anda dapat menerapkan proses peninjauan yang sama untuk basis data dan sumber data lain yang menggunakan autentikasi IAM, seperti SQS atau penyimpanan data pihak ketiga.
4. Gunakan AWS IAM Access Analyzer:

- Anda dapat mengonfigurasi AWS IAM Access Analyzer untuk menganalisis bucket Amazon S3 dan menghasilkan temuan saat sebuah kebijakan S3 memberikan akses ke entitas eksternal.

5. Terapkan mekanisme penentuan versi dan penguncian objek:
   - Gunakan penentuan versi Amazon S3 untuk mempertahankan versi objek sebelumnya, sehingga menyediakan pemulihan dari penghapusan atau penimpaan yang tidak disengaja.
   - Gunakan Kunci Objek Amazon S3 untuk menyediakan kontrol akses wajib untuk objek, sehingga mencegahnya dihapus atau ditimpa, bahkan oleh pengguna root, hingga kunci ini kedaluwarsa.
   - Gunakan Kunci Vault Amazon S3 Glacier untuk arsip yang disimpan di Amazon S3 Glacier.
6. Gunakan Inventaris Amazon S3:
   - Gunakan Inventaris Amazon S3 untuk mengaudit dan melaporkan replikasi dan status enkripsi objek S3 Anda.
7. Tinjau izin berbagi Amazon EBS dan AMI:
   - Tinjau izin berbagi Anda untuk Amazon EBS dan berbagi AMI untuk memastikan image dan volume Anda tidak dibagikan ke Akun AWS di luar beban kerja Anda.
8. Tinjau Pembagian AWS Resource Access Manager secara berkala:
   - Anda dapat menggunakan AWS Resource Access Manager untuk membagikan sumber daya, seperti kebijakan AWS Network Firewall, aturan Amazon Route 53 Resolver, dan subnet dalam Amazon VPC Anda.
   - Lakukan audit terhadap sumber daya yang dibagikan secara rutin dan hentikan pembagian sumber daya yang sudah tidak perlu dibagikan.

Sumber daya

Praktik-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 Anda
- Gambaran umum mengenai pengelolaan akses ke sumber daya AWS KMS Anda

- Aturan AWS Config
- Amazon S3 + Amazon CloudFront: Pertandingan yang Dibuat di Cloud
- Menggunakan penentuan versi
- Mengunci Objek Menggunakan Kunci Objek Amazon S3
- Berbagi Snapshot Amazon EBS
- AMI Bersama
- Meng-host aplikasi satu halaman di Amazon S3
- Kunci Kondisi Global AWS
- Membangun Perimeter Data di AWS

Video terkait:

- Mengamankan Penyimpanan Blok di 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.

Mengamankan data dari antara VPC atau lokasi on-premise: Anda dapat menggunakan AWS PrivateLinkuntuk membuat sebuah sambungan jaringan privat dan aman antara Amazon Virtual Private Cloud (Amazon VPC) atau konektivitas on-premise 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 Autentikasikan 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 akan mencegah pengungkapan materi kunci privat sertifikat dan secara otomatis memperpanjang sertifikat secara berkala. Mekanisme ini juga terintegrasi dengan layanan-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.

Anti-pola 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 dengan menggunakan sertifikat TLS
- Peningkatan keamanan dan keterauditan tindakan-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 diterapkan: Tinggi

Panduan implementasi

Beban kerja modern banyak memanfaatkan komunikasi jaringan terenkripsi dengan 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 AWS publik maupun privat. ACM mengeluarkan sertifikat privat dengan menggunakan AWS Private CA dan terintegrasi dengan banyak layanan terkelola AWS lainnya untuk menyediakan sertifikat TLS yang aman untuk beban kerja. ACM juga dapat mengeluarkan sertifikat tepercaya publik dari Amazon Trust Services. Sertifikat publik dari ACM dapat digunakan pada beban kerja sisi publik karena browser dan sistem operasi modern memercayai sertifikat ini secara default.

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 menyediakan dukungan khusus untuk penyediaan, pengelolaan, dan deployment sertifikat PKI ke perangkat IoT. AWS IoT Core menyediakan mekanisme khusus untuk melakukan onboarding perangkat IoT ke infrastruktur kunci publik Anda dalam skala besar.

Beberapa layanan AWS, seperti Amazon API Gateway dan Elastic Load Balancing, menawarkan kemampuannya sendiri untuk menggunakan sertifikat guna mengamankan koneksi aplikasi. Misalnya, API Gateway dan Penyeimbang Beban Aplikasi (ALB) mendukung TLS mutual (mTLS) menggunakan sertifikat klien yang Anda buat dan ekspor menggunakan AWS Management Console, CLI, atau API.

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, buatlah 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 para 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-langkah implementasi

1. Tentukan layanan-layanan AWS yang 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 mengeluarkan 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.
    - Pertimbangkan untuk menggunakan fungsionalitas mTLS native dalam layanan seperti Amazon API Gateway atau Penyeimbang Beban Aplikasi.
2. Implementasikan perpanjangan sertifikat otomatis jika memungkinkan:
    - Gunakan pembaruan terkelola ACM untuk sertifikat yang diterbitkan oleh ACM bersama dengan layanan terkelola AWS yang terintegrasi.
3. Bangun jejak pencatatan log dan jejak audit:

- Aktifkan log CloudTrail untuk melacak akses ke akun yang memiliki otoritas sertifikat. Pertimbangkan untuk mengonfigurasi validasi integritas file log di CloudTrail untuk memverifikasi keaslian data log.
- Buat dan tinjau secara berkala laporan audit yang mencantumkan sertifikat yang telah dikeluarkan atau dicabut oleh CA privat Anda. Laporan-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 membaca panduan mengenai cara mengonfigurasi bucket S3 ini berdasarkan persyaratan beban kerja Anda, silakan lihat Merencanakan daftar pencabutan sertifikat (CRL).

Sumber daya

Praktik-praktik terbaik terkait:

- SEC02-BP02 Menggunakan kredensial sementara
- SEC08-BP01 Mengimplementasikan manajemen kunci yang aman
- SEC09-BP03 Autentikasikan komunikasi jaringan

Dokumen terkait:

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

Berlakukan persyaratan-persyaratan enkripsi yang Anda tetapkan berdasarkan kebijakan, kewajiban berdasarkan 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 akan membantu menjaga kerahasiaan data, bahkan ketika data berada di jaringan yang tidak tepercaya.

Hasil yang diinginkan: Anda mengenkripsi lalu lintas jaringan antara sumber daya Anda dan internet untuk mengurangi akses tidak sah ke data. Anda mengenkripsi lalu lintas jaringan dalam lingkungan AWS internal Anda sesuai dengan persyaratan keamanan Anda. Anda mengenkripsi data bergerak menggunakan protokol TLS aman dan cipher suite.

Anti-pola 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 diterapkan: Tinggi

Panduan implementasi

Layanan-layanan AWS menyediakan titik akhir HTTPS menggunakan TLS untuk komunikasi, memberikan enkripsi data bergerak saat berkomunikasi dengan API AWS. Protokol HTTP yang tidak aman dapat diaudit dan diblokir di Cloud Privat Virtual (VPC) melalui penggunaan grup keamanan. Permintaan HTTP juga dapat secara otomatis dialihkan ke HTTPS di Amazon CloudFront atau di Penyeimbang Beban Aplikasi. Anda dapat menggunakan kebijakan bucket Amazon Simple Storage Service (Amazon S3) untuk membatasi kemampuan mengunggah objek melalui HTTP, yang secara efektif menerapkan penggunaan HTTPS untuk pengunggahan objek ke bucket Anda. 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 memudahkan enkripsi lalu lintas. Verifikasikan bahwa klien Anda melakukan panggilan ke API AWS menggunakan setidaknya TLS 1.2 karena AWS telah menghentikan penggunaan versi TLS sebelumnya per Februari 2024. Kami menyarankan Anda menggunakan TLS 1.3. Jika Anda memiliki persyaratan khusus untuk enkripsi bergerak, Anda dapat menemukan solusi pihak ketiga yang tersedia di AWS Marketplace.

Langkah-langkah implementasi

- Terapkan enkripsi data bergerak: Persyaratan enkripsi yang Anda tetapkan harus didasarkan pada standar dan praktik terbaik paling baru 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 sambungan eksternal: Pertimbangkan penggunaan VPN IPsec untuk mengamankan sambungan titik ke titik atau jaringan ke jaringan untuk membantu menyediakan privasi sekaligus integritas data.
- Konfigurasikan protokol aman di penyeimbang beban: Pilih sebuah kebijakan keamanan yang menyediakan cipher suite terkuat yang didukung oleh klien yang akan terhubung ke pendengar. Buat pendengar HTTPS untuk Penyeimbang Beban Aplikasi Anda.
- Konfigurasikan protokol yang aman di Amazon Redshift:: Konfigurasikan klaster Anda agar mewajibkan koneksi lapisan soket aman (SSL) atau keamanan lapisan pengangkutan (TLS).
- Konfigurasikan protokol yang aman: Tinjau dokumentasi layanan AWS untuk menentukan kemampuan enkripsi bergerak.
- Konfigurasikan akses yang aman saat mengunggah ke bucket Amazon S3: Gunakan kontrol kebijakan bucket Amazon S3 untuk menerapkan akses yang aman ke data.
- Pertimbangkan untuk menggunakan AWS Certificate Manager: ACM akan memungkinkan Anda untuk menyediakan, mengelola, dan men-deploy sertifikat TLS publik untuk digunakan dengan layanan AWS.
- Pertimbangkan untuk menggunakan AWS Private Certificate Authority untuk kebutuhan PKI privat: AWS Private CA akan memungkinkan Anda membuat hierarki otoritas sertifikat pribadi (CA) untuk mengeluarkan sertifikat X.509 entitas akhir yang dapat digunakan untuk membuat saluran TLS terenkripsi.

Sumber daya

Dokumen terkait:

- Menggunakan HTTPS dengan CloudFront
- Hubungkan VPC Anda ke jaringan jarak jauh menggunakan AWS Virtual Private Network
- Buat pendengar HTTPS untuk Penyeimbang Beban Aplikasi 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 Autentikasikan komunikasi jaringan

Verifikasikan identitas komunikasi 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 alur jaringan dan mengurangi dampak akses yang tidak sah.

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

Anti-pola umum:

- Arus 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.
- Arus lalu lintas yang terlalu permisif antara komponen layanan atau sumber daya lain di VPC.

Manfaat menjalankan praktik terbaik ini:

- Membatasi cakupan dampak untuk akses tidak sah ke satu bagian dari beban kerja.

- Memberikan tingkat jaminan yang lebih tinggi bahwa tindakan hanya dilakukan oleh entitas-entitas yang diautentikasi.
- Meningkatkan pemisahan layanan dengan menentukan dan menerapkan antarmuka transfer data yang diinginkan secara jelas.
- Meningkatkan pemantauan, pembuatan log, dan respons insiden melalui atribusi permintaan dan antarmuka komunikasi yang ditentukan 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 diterapkan: Rendah

Panduan implementasi

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

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

Mengenkripsi lalu lintas utara-selatan adalah praktik yang umum, sedangkan 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 batasan jaringan yang sama, sebaiknya enkripsi, autentikasi, dan otorisasi komunikasi di antara layanan-layanan tersebut tetap dilakukan, itulah praktik terbaiknya.

Sebagai contoh, API layanan AWS menggunakan protokol tanda tangan AWS Signature Version 4 (SigV4) untuk mengautentikasi pemanggil, tidak peduli dari jaringan mana permintaan itu berasal. Autentikasi ini memastikan bahwa API AWS dapat memverifikasi identitas yang meminta tindakan, dan identitas tersebut kemudian dapat digabungkan dengan kebijakan untuk membuat sebuah keputusan otorisasi guna menentukan apakah tindakan tersebut harus diizinkan atau tidak.

Layanan-layanan seperti Amazon VPC Lattice dan Amazon API Gateway akan memungkinkan Anda untuk menggunakan protokol tanda tangan SigV4 yang sama untuk menambahkan autentikasi dan otorisasi ke lalu lintas timur-barat yang ada di 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 dengan 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 dengan menggunakan 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 untuk menyediakan autentikasi mTLS untuk komunikasi antar atau intra-beban kerja. Penyeimbang Beban Aplikasi juga mendukung mTLS untuk beban kerja internal atau eksternal. 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 mengendalikan akses ke layanan oleh pengguna, tetapi sekarang keduanya menjadi populer untuk lalu lintas layanan-ke-layanan juga. API Gateway menyediakan pemberi otorisasi JSON Web Token (JWT), yang memungkinkan beban kerja membatasi akses ke rute API dengan menggunakan JWT yang dikeluarkan dari penyedia identitas OIDC atau OAuth 2.0. Cakupan OAuth2 dapat digunakan sebagai sebuah 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-langkah implementasi

- Tentukan dan dokumentasikan alur jaringan beban kerja Anda: Langkah pertama yang harus dilakukan untuk menerapkan strategi pertahanan mendalam adalah menentukan arus lalu lintas beban kerja Anda.
  - Buatlah sebuah diagram alur data yang secara jelas menentukan bagaimana data ditransmisikan antara berbagai layanan yang membentuk beban kerja Anda. Diagram ini merupakan langkah pertama untuk menerapkan alur-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 alir data juga dapat berguna saat Anda melakukan latihan pemodelan ancaman, seperti yang dijelaskan dalam SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi dengan menggunakan sebuah model ancaman.

- Tetapkan kontrol jaringan: Pertimbangkan kemampuan AWS untuk membuat kontrol jaringan yang selaras dengan aliran data Anda. Meskipun batasan-batasan jaringan seharusnya tidak menjadi satu-satunya kontrol keamanan, batasan-batasan tersebut menyediakan lapisan pada strategi pertahanan mendalam untuk melindungi beban kerja Anda.
  - Gunakan grup keamanan untuk menetapkan definisi dan membatasi aliran data antar sumber daya.
  - Pertimbangkan untuk menggunakan AWS PrivateLink untuk berkomunikasi dengan layanan AWS dan layanan pihak ketiga yang mendukung AWS PrivateLink. Data yang dikirim melalui suatu titik akhir antarmuka AWS PrivateLink tetap berada di dalam tulang punggung jaringan AWS dan tidak melintasi Internet publik.
- Menerapkan autentikasi dan otorisasi di seluruh layanan dalam beban kerja Anda: Pilih rangkaian layanan AWS yang paling tepat untuk menyediakan arus lalu lintas terautentikasi dan terenkripsi dalam beban kerja Anda.
  - Pertimbangkan Amazon VPC Lattice untuk mengamankan komunikasi layanan-ke-layanan. VPC Lattice dapat menggunakan autentikasi SigV4 yang dikombinasikan dengan kebijakan autentikasi untuk mengontrol akses layanan-ke-layanan.
  - Untuk komunikasi layanan-ke-layanan menggunakan mTLS, pertimbangkan API Gateway atau Penyeimbang Beban Aplikasi. AWS Private CA dapat digunakan untuk membuat hierarki CA privat yang mampu mengeluarkan sertifikat untuk digunakan dengan mTLS.
  - Saat mengintegrasikan dengan layanan menggunakan OAuth 2.0 atau OIDC, pertimbangkan API Gateway yang menggunakan pemberi otorisasi JWT.
  - Untuk komunikasi antara beban kerja Anda dan perangkat IoT, pertimbangkan untuk menggunakan AWS IoT Core, yang menyediakan beberapa opsi untuk enkripsi dan autentikasi lalu lintas jaringan.
- Memantau akses yang tidak sah: Lakukan pemantauan secara terus-menerus terhadap saluran komunikasi yang tidak diinginkan, principal yang tidak sah yang mencoba mengakses sumber daya yang dilindungi, dan pola akses yang tidak tepat lainnya.
  - Jika Anda menggunakan VPC Lattice 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 aliran VPC untuk merekam metadata pada alur jaringan dan meninjau anomali secara berkala.

- Lihat Panduan Respons Insiden Keamanan AWS dan bagian Respons Insiden pilar keamanan Kerangka Kerja AWS Well-Architected untuk membaca panduan lebih lanjut tentang perencanaan, simulasi, dan penanggulangan insiden keamanan.

Sumber daya

Praktik-praktik terbaik terkait:

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

Dokumen terkait:

- Mengevaluasi metode kontrol akses untuk mengamankan API Amazon API Gateway
- Mengonfigurasi autentikasi TLS timbal balik untuk API REST
- Cara mengamankan titik akhir HTTP API Gateway dengan pemberi otorisasi JWT
- Mengotorisasi panggilan langsung ke layanan AWS menggunakan penyedia kredensial AWS IoT Core
- Panduan Respons Insiden Keamanan AWS

Video terkait:

- AWS re:Invent 2022: Memperkenalkan Kisi VPC
- AWS re:Invent 2020: Autentikasi API nirserver untuk API HTTP di AWS

Contoh terkait:

- Lokakarya Amazon VPC Lattice
- Zero-Trust Episode 1 – Lokakarya Perimeter Layanan Phantom

Respons insiden

Dengan kontrol detektif dan preventif yang matang sekalipun, organisasi Anda harus mengimplementasikan mekanisme untuk memberikan respons dan melakukan mitigasi atas 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-aspek respons insiden AWS

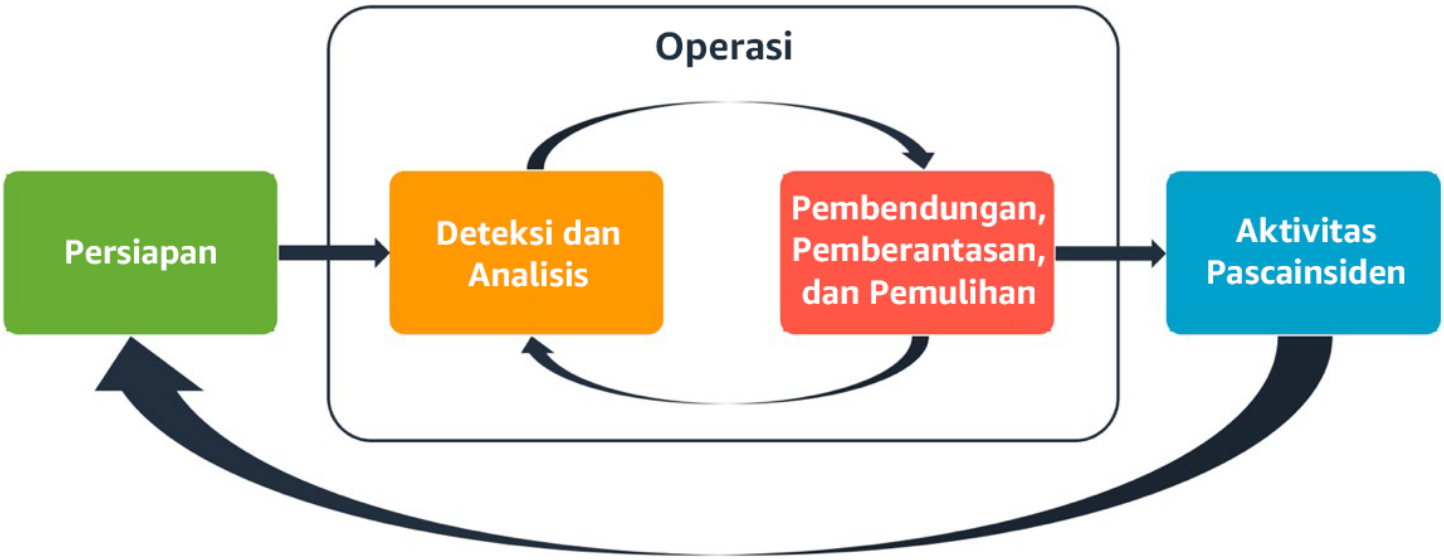
Semua pengguna AWS dalam suatu organisasi harus memiliki pemahaman dasar tentang proses respons insiden keamanan, dan staf keamanan harus memahami bagaimana merespons masalah keamanan. Pendidikan, pelatihan, dan pengalaman sangat penting agar program respons insiden cloud berjalan dengan baik, dan idealnya diimplementasikan dengan baik sebelum harus menangani kemungkinan insiden keamanan. Fondasi program respons insiden yang baik di cloud adalah Persiapan, Operasi, dan Aktivitas Pasca Insiden.

Untuk memahami setiap aspek ini, lihat deskripsi berikut:

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

- Aktivitas pasca insiden: Lakukan iterasi pada hasil simulasi dan peristiwa keamanan Anda untuk meningkatkan efektivitas respons Anda, sehingga respons dan investigasi yang dilakukan bisa lebih bernilai, dan mengurangi risiko lebih lanjut. Anda harus belajar dari insiden dan memiliki sikap kepemilikan yang kuat terhadap 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-aspek respons insiden AWS

Tujuan desain respons cloud

Meskipun proses umum dan mekanisme respons insiden sebagaimana didefinisikan di dalam NIST SP 800-61 Computer Security Incident Handling Guide, masih benar, kami mendorong Anda untuk mengevaluasi tujuan-tujuan desain spesifik ini, yang relevan untuk merespons insiden keamanan di lingkungan cloud:

- Menetapkan tujuan respons: Bekerja sama dengan pemangku kepentingan, penasihat hukum, dan kepemimpinan organisasi untuk menentukan tujuan dalam merespons suatu 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.
- Merespons menggunakan cloud: Menerapkan pola respons di dalam cloud, tempat peristiwa dan data terjadi.

- 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 khusus untuk respons. Gunakan tag, metadata, dan mekanisme yang menerapkan kebijakan retensi. Anda harus memahami layanan apa yang Anda gunakan, lalu mengidentifikasi persyaratan untuk menginvestigasi layanan tersebut. Untuk membantu memahami lingkungan Anda, Anda juga dapat menggunakan pemberian tag.
- Gunakan mekanisme deployment ulang: Jika anomali keamanan dapat dikaitkan dengan kesalahan konfigurasi, remediasinya mungkin cukup dengan menghapus perbedaan konfigurasi ini dengan deployment ulang sumber daya menggunakan konfigurasi yang tepat. Jika kemungkinan gangguan teridentifikasi, pastikan deployment ulang Anda mencakup mitigasi akar masalah yang berhasil dan diverifikasi.
- Otomatiskan jika memungkinkan: Ketika masalah muncul atau insiden berulang, bangun mekanisme untuk melakukan triase secara terprogram dan merespons peristiwa umum. Gunakan respons manusia untuk insiden yang unik, kompleks, atau sensitif yang tidak cukup ditangani oleh otomatisasi.
- Pilih solusi yang dapat diskalakan: Berusahalah untuk mengimbangi skalabilitas pendekatan organisasi Anda terhadap komputasi cloud. Implementasikan mekanisme deteksi dan respons yang diskalakan di seluruh lingkungan Anda untuk secara efektif mengurangi waktu antara deteksi dan respons.
- Pelajari dan tingkatkan proses Anda: Bersikaplah proaktif ketika mengidentifikasi kesenjangan dalam proses, alat, atau orang Anda, dan terapkan rencana untuk memperbaikinya. Simulasi merupakan metode yang aman untuk menemukan celah dan menyempurnakan proses.

Sasaran desain ini merupakan pengingat untuk meninjau implementasi arsitektur Anda agar dapat melakukan respons insiden dan deteksi ancaman. Saat Anda merencanakan implementasi cloud Anda, pikirkan tentang merespons 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 ini harus tersedia bagi responden insiden melalui alur deployment. Alat dan fungsi tersebut tidak boleh statis karena dapat menyebabkan risiko yang lebih besar.

Persiapan

Persiapan untuk menghadapi insiden merupakan hal yang sangat penting agar respons insiden bisa dilakukan dengan cepat dan efektif. Persiapan dilakukan di tiga domain:

- Orang: Dalam mempersiapkan orang-orang Anda untuk menghadapi insiden keamanan, pemangku kepentingan yang relevan perlu diidentifikasi untuk respons insiden, dan dilatih tentang respons insiden dan teknologi cloud.
- Proses: Dalam mempersiapkan proses Anda untuk menghadapi insiden keamanan, perlu adanya pendokumentasian arsitektur, pengembangan rencana respons insiden menyeluruh, dan pembuatan playbook agar respons terhadap peristiwa keamanan bisa dilakukan secara konsisten.
- Teknologi: Dalam mempersiapkan teknologi Anda untuk menghadapi insiden keamanan, perlu adanya pengaturan akses, agregasi dan pemantauan log yang diperlukan, penerapan mekanisme peringatan yang efektif, dan pengembangan respons serta kemampuan penyelidikan.

Setiap domain ini sama pentingnya agar respons insiden berjalan efektif. Tanpa ketiga domain ini, program respons insiden tidak akan lengkap atau efektif. Anda perlu mempersiapkan orang, proses, dan teknologi dengan integrasi yang erat agar siap menghadapi suatu insiden.

Praktik terbaik

- SEC10-BP01 Identifikasikan sumber daya eksternal dan personel penting
- SEC10-BP02 Membuat rencana manajemen insiden
- SEC10-BP03 Siapkan 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 penting

Identifikasikan personel, sumber daya, dan kewajiban hukum internal serta eksternal untuk membantu organisasi Anda merespons insiden.

Hasil yang diinginkan: Anda memiliki daftar personel kunci, informasi kontak mereka, dan peran yang mereka mainkan saat menanggapi peristiwa keamanan yang terjadi. Anda meninjau informasi ini secara rutin dan memperbaruinya untuk menyesuaikan dengan perubahan personel dari perspektif alat internal dan eksternal. Anda mempertimbangkan semua penyedia dan vendor layanan pihak ketiga saat mendokumentasikan informasi ini, termasuk partner keamanan, penyedia cloud, dan aplikasi perangkat lunak sebagai layanan (SaaS). Saat peristiwa keamanan berlangsung, tersedia

personel dengan tingkat tanggung jawab, konteks, dan akses yang sesuai untuk melakukan respons dan pemulihan.

Anti-pola umum:

- Tidak memelihara daftar terbaru personel penting dengan informasi kontak, peran mereka, dan tanggung jawab mereka saat merespons peristiwa keamanan.
- Mengasumsikan bahwa setiap orang mengetahui staf yang bertanggung jawab, dependensi, infrastruktur, dan solusi ketika melakukan respons dan pemulihan dari suatu peristiwa.
- Tidak memiliki repositori dokumen atau pengetahuan yang merepresentasikan desain infrastruktur atau aplikasi utama.
- Tidak memiliki proses onboarding yang tepat bagi karyawan baru untuk berkontribusi secara efektif terhadap respons peristiwa keamanan, seperti melakukan simulasi peristiwa.
- Tidak memiliki jalur eskalasi ketika personel penting tidak tersedia untuk sementara waktu atau tidak merespons saat terjadi peristiwa keamanan.

Manfaat menjalankan praktik terbaik ini: Praktik ini akan mengurangi triase dan waktu respons yang dihabiskan untuk mengidentifikasi personel yang tepat dan peran mereka selama suatu peristiwa. Minimalkan waktu yang terbuang saat terjadi sebuah peristiwa dengan memelihara daftar terbaru personel penting dan peran mereka sehingga Anda dapat mendatangkan orang-orang yang tepat untuk melakukan triase dan pemulihan dari sebuah peristiwa.

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

Panduan implementasi

Identifikasikan personel kunci dalam organisasi Anda: Kelola daftar kontak personel di dalam organisasi Anda yang perlu Anda libatkan. Tinjau dan perbarui informasi ini secara rutin jika terjadi perpindahan personel, seperti perubahan organisasi, promosi, dan perubahan tim. Hal ini penting terutama untuk peran penting seperti manajer insiden, responden insiden, dan kepala komunikasi.

- Manajer insiden: Manajer insiden memiliki otoritas keseluruhan selama merespons peristiwa.
- Responden insiden: Responden insiden bertanggung jawab atas kegiatan investigasi dan remediasi. Orang-orang ini dapat berbeda berdasarkan jenis peristiwanya, tetapi biasanya adalah developer dan tim operasi yang bertanggung jawab atas aplikasi yang terkena dampak.
- Pimpinan komunikasi: Pimpinan komunikasi bertanggung jawab atas komunikasi internal dan eksternal, terutama komunikasi dengan lembaga publik, regulator, dan pelanggan.

- Proses orientasi: Secara teratur melakukan pelatihan dan orientasi karyawan baru untuk membekali mereka dengan keterampilan dan pengetahuan yang diperlukan guna berkontribusi secara efektif terhadap upaya respons insiden. Terapkan simulasi dan latihan praktik langsung sebagai bagian dari proses orientasi untuk memfasilitasi kesiapan mereka
- Ahli bidang studi (SME): Dalam kasus tim terdistribusi dan otonom, kami sarankan Anda mengidentifikasi SME untuk beban kerja kritis misi. Mereka memberikan wawasan tentang operasi dan klasifikasi data pada beban kerja krusial yang terpengaruh dalam peristiwa.

Contoh format tabel:

```
  | Role | Name | Contact Information | Responsibilities |
1 | ——- | ——- | ——- | ——- |
2 | Incident Manager | Jane Doe| jane.doe@example.com | Overall authority during
 response |
3 | Incident Responder | John Smith | john.smith@example.com | Investigation and
 remediation |
4 | Communications Lead | Emily Johnson | emily.johnson@example.com | Internal and
 external communications |
5 | Communications Lead | Michael Brown | michael.brown@example.com | Insights on
 critical workloads |
```

Pertimbangkan untuk menggunakan fitur AWS Systems Manager Incident Manager untuk merekam kontak utama, menentukan rencana respons, mengotomatiskan jadwal panggilan, dan membuat rencana eskalasi. Lakukan otomatisasi dan rotasi semua staf melalui jadwal jaga, sehingga tanggung jawab atas beban kerja dibagi di antara pemiliknya. Hal ini mendukung praktik-praktik yang baik, seperti menghasilkan metrik dan log yang relevan serta menentukan ambang batas alarm yang penting untuk beban kerja.

Identifikasi mitra eksternal: Perusahaan menggunakan alat yang dibangun oleh vendor perangkat lunak independen (ISV), mitra, dan subkontraktor untuk membangun solusi yang memiliki diferensiasi bagi pelanggan mereka. Libatkan personel penting dari pihak-pihak ini yang dapat membantu merespons dan melakukan pemulihan dari suatu insiden. Sebaiknya Anda mendaftar untuk tingkat Dukungan yang sesuai untuk mendapatkan akses cepat ke ahli bidang studi AWS melalui kasus dukungan. Pertimbangkan untuk melakukan hal yang serupa dengan semua penyedia solusi yang krusial untuk beban kerja. Beberapa peristiwa keamanan tertentu mengharuskan bisnis yang terdaftar di bursa saham memberi tahu lembaga publik dan badan pengatur yang relevan tentang peristiwa yang terjadi dan dampaknya. Pelihara dan perbarui informasi kontak untuk departemen yang relevan dan orang-orang yang bertanggung jawab.

Langkah-langkah implementasi

1. Siapkan sebuah solusi manajemen insiden.
   a. Pertimbangkan untuk melakukan deployment Incident Manager di akun Security Tooling Anda.
2. Tentukan kontak dalam solusi manajemen insiden Anda.
   a. Tentukan minimal dua jenis saluran kontak untuk setiap kontak (seperti SMS, telepon, atau email) guna memastikan kontak tersebut dapat dihubungi saat insiden berlangsung.
3. Tentukan rencana respons.
   a. Identifikasi kontak yang paling tepat untuk digunakan selama insiden. Tentukan rencana eskalasi yang selaras dengan peran yang dimiliki oleh personel yang akan dilibatkan, bukan kontak individu. Pertimbangkan untuk memasukkan kontak yang mungkin bertanggung jawab untuk memberi tahu entitas eksternal, meskipun mereka tidak terlibat langsung untuk menyelesaikan insiden.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP03 Aktivitas operasi memiliki pemilik teridentifikasi yang bertanggung jawab atas kinerjanya

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS

Contoh terkait:

- Kerangka kerja playbook pelanggan AWS
- Bersiap dan merespons insiden keamanan di lingkungan AWS Anda

Alat terkait:

- Peluncuran AWS Systems Manager Incident Manager

Video terkait:

- Pendekatan Amazon terhadap keamanan selama pengembangan

SEC10-BP02 Membuat rencana manajemen insiden

Dokumen pertama yang dikembangkan untuk respons insiden adalah rencana respons insiden. Rencana respons insiden dirancang untuk menjadi dasar 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 terukur. Ketika sebuah peristiwa keamanan terjadi, langkah dan alur kerja yang jelas dapat membantu Anda merespons secara tepat waktu. Anda mungkin sudah memiliki proses respons insiden sendiri. Terlepas dari keadaan saat ini, penting untuk memperbarui, mengulangi, dan menguji proses respons insiden Anda secara teratur.

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

Panduan implementasi

Rencana manajemen insiden sangat penting untuk merespons, memitigasi, dan pulih dari potensi dampak yang ditimbulkan insiden keamanan. Rencana manajemen insiden adalah sebuah 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 Anda membuat sebuah rencana manajemen insiden, Anda harus 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 mematuhi FedRAMP di Amerika Serikat, ikuti rekomendasi dalam NIST SP 800-61 Panduan Penanganan Keamanan Komputer. Demikian pula, ketika Anda mengoperasikan beban kerja yang menyimpan informasi pengenal pribadi (PII), pertimbangkan cara melindungi dan merespons masalah yang terkait dengan residensi dan penggunaan data.

Saat membuat rencana manajemen insiden untuk beban kerja Anda di AWS, mulailah dengan Model Tanggung Jawab Bersama AWS untuk membangun pendekatan pertahanan mendalam terhadap respons insiden. Dalam model ini, AWS mengelola keamanan cloud, dan Anda bertanggung jawab atas keamanan di cloud tersebut. 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 yang berorientasi cloud.

Rencana manajemen insiden yang efektif harus diulang-ulang (iterasi) secara berkelanjutan, dan harus tetap mutakhir sesuai tujuan-tujuan operasi cloud Anda. Pertimbangkan untuk menggunakan rencana implementasi yang diuraikan di bawah ini saat Anda membuat dan mengembangkan rencana manajemen insiden Anda.

Langkah-langkah implementasi

1. Tentukan peran dan tanggung jawab dalam organisasi Anda untuk menangani peristiwa keamanan. Hal ini harus melibatkan perwakilan dari berbagai departemen, termasuk:
   - Sumber daya manusia (SDM)
   - Tim eksekutif
   - Departemen hukum
   - Pemilik dan developer aplikasi (ahli bidang studi, atau SME)
2. Uraikan dengan jelas siapa yang mengemban tanggung jawab, memegang akuntabilitas, dimintai pertimbangan, dan diberi informasi atau Responsible, Accountable, Consulted, and Informed (RACI) selama suatu insiden. Buat bagan RACI untuk memfasilitasi komunikasi yang cepat dan langsung, serta menguraikan kepemimpinan di berbagai tahap peristiwa dengan jelas.
3. Libatkan pemilik aplikasi dan developer (SME) selama insiden, karena mereka dapat memberikan informasi dan konteks yang berharga untuk membantu mengukur dampaknya. Bangun hubungan dengan SME ini, dan lakukan latihan skenario respons insiden dengan mereka sebelum insiden sebenarnya terjadi.
4. Libatkan mitra tepercaya atau ahli eksternal dalam proses investigasi atau respons karena mereka dapat memberikan keahlian dan perspektif tambahan.
5. Selaraskan rencana dan peran manajemen insiden Anda dengan peraturan setempat atau persyaratan kepatuhan yang mengatur organisasi Anda.
6. Latih dan uji rencana respons insiden Anda secara teratur, dan libatkan semua peran dan tanggung jawab yang ditentukan. Tindakan ini membantu merampingkan proses dan memverifikasi bahwa Anda memiliki respons yang terkoordinasi dan efisien terhadap insiden keamanan.
7. Tinjau serta perbarui peran, tanggung jawab, dan bagan RACI secara berkala, atau ketika struktur atau persyaratan organisasi Anda berubah.

Memahami tim respons dan dukungan AWS

- AWS Dukungan
  - Dukungan menawarkan berbagai rencana yang menyediakan akses ke alat dan keahlian yang mendukung keberhasilan dan kesehatan operasional solusi AWS. Jika Anda memerlukan dukungan teknis dan sumber daya lainnya untuk membantu merencanakan, melakukan deployment, dan mengoptimalkan lingkungan AWS, Anda dapat memilih paket dukungan yang paling sesuai dengan kasus penggunaan AWS Anda.
  - Pertimbangkan Pusat Dukungan di AWS Management Console (perlu masuk ke akun) sebagai titik kontak utama guna mendapatkan dukungan untuk masalah-masalah yang memengaruhi sumber daya AWS Anda. Akses ke Dukungan dikontrol oleh AWS Identity and Access Management. Untuk informasi selengkapnya tentang mendapatkan akses ke fitur-fitur dukungan Dukungan, silakan lihat Memulai dengan Dukungan.
- Tim Respons Insiden Pelanggan (CIRT) AWS
  - Tim Respons Insiden Pelanggan (CIRT) AWS adalah tim AWS global khusus yang selalu siap untuk, 24 jam sehari, 7 hari seminggu, memberikan dukungan kepada pelanggan selama terjadinya peristiwa keamanan aktif di sisi pelanggan dalam Model Tanggung Jawab Bersama AWS.
  - Ketika CIRT AWS mendukung Anda, mereka memberikan bantuan dengan melakukan evaluasi awal (triase) 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 Dukungan.
- Dukungan respons DDoS
  - AWS menawarkan AWS Shield, yang menyediakan layanan perlindungan penolakan layanan terdistribusi terkelola (DDoS) yang akan melindungi aplikasi web yang berjalan di AWS. Shield menyediakan deteksi yang selalu aktif dan mitigasi integral otomatis yang dapat meminimalkan waktu henti dan latensi aplikasi, sehingga tidak perlu melibatkan Dukungan untuk mendapatkan manfaat dari perlindungan DDoS. Terdapat dua tingkatan Shield: AWS Shield Standard dan AWS Shield Advanced. Untuk mengetahui perbedaan antara kedua tingkatan ini, silakan lihat Dokumentasi fitur Shield.
- AWS Managed Services (AMS)
  - AWS Managed Services (AMS) menyediakan pengelolaan infrastruktur AWS yang berkelanjutan, sehingga Anda dapat fokus pada aplikasi Anda. Dengan menerapkan praktik terbaik untuk

memelihara infrastruktur Anda, AMS membantu mengurangi biaya operasional dan risiko Anda. AMS mengotomatiskan aktivitas 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 memberikan respons baris pertama 24/7 terhadap peringatan. Saat peringatan dimulai, AMS mengikuti seperangkat standar playbook otomatis dan manual untuk memverifikasi respons yang konsisten. Playbook ini dibagikan kepada pelanggan AMS saat orientasi agar mereka dapat mengembangkan dan mengoordinasikan respons dengan AMS.

Kembangkan rencana respons insiden

Rencana respons insiden dirancang untuk menjadi dasar bagi program dan strategi respons insiden Anda. Rencana respons insiden harus dalam bentuk 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: Membuat daftar pemangku kepentingan respons insiden dan menjabarkan peran mereka ketika insiden terjadi.
- Rencana komunikasi: Detail informasi kontak dan bagaimana mekanisme komunikasi selama insiden.
- Buat pencadangan metode komunikasi: Memiliki komunikasi alternatif terpisah sebagai cadangan untuk komunikasi insiden merupakan praktik terbaik. Contoh aplikasi yang menyediakan saluran komunikasi out-of-band yang aman adalah AWS Wickr.
- Fase respons insiden dan tindakan yang perlu diambil: Mengenumerasi fase respons insiden, (misalnya, mendeteksi, menganalisis, memberantas, menahan, dan memulihkan), termasuk tindakan tingkat tinggi yang harus diambil dalam fase-fase tersebut.
- Definisi keparahan insiden dan prioritas: Memerinci cara mengklasifikasikan tingkat keparahan suatu insiden, bagaimana memprioritaskan insiden, lalu bagaimana definisi keparahan mempengaruhi prosedur eskalasi.

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

Sumber daya

Praktik-praktik terbaik terkait:

- SEC04 Deteksi

Dokumen terkait:

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

SEC10-BP03 Siapkan kemampuan forensik

Menjelang insiden keamanan, pertimbangkan untuk mengembangkan kemampuan forensik guna mendukung investigasi peristiwa keamanan.

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

Konsep dari forensik lokal tradisional berlaku untuk. AWS Untuk informasi kunci untuk mulai membangun kemampuan forensik di AWS Cloud, lihat Strategi lingkungan investigasi forensik di. AWS Cloud

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

- Koleksi: Kumpulkan AWS log yang relevan, seperti AWS CloudTrail, AWS Config, Log VPC Aliran, dan log tingkat host. Kumpulkan snapshot, backup, dan dump memori dari sumber daya yang terkena dampak AWS jika tersedia.
- Pemeriksaan: Memeriksa data yang dikumpulkan dengan mengekstraksi dan menilai informasi yang relevan.
- Analisis: Menganalisis data yang dikumpulkan untuk memahami insiden dan menarik kesimpulan dari insiden tersebut.
- Pelaporan: Menyajikan informasi yang dihasilkan dari fase analisis.

Langkah-langkah implementasi

Persiapkan lingkungan forensik Anda

AWS Organizationsmembantu Anda mengelola dan mengatur AWS lingkungan secara terpusat saat Anda tumbuh dan meningkatkan AWS sumber daya. Sebuah AWS organisasi mengkonsolidasikan Anda Akun AWS sehingga Anda dapat mengelolanya sebagai satu unit. Anda dapat menggunakan unit organisasi (OUs) untuk mengelompokkan akun bersama untuk mengelola sebagai satu unit.

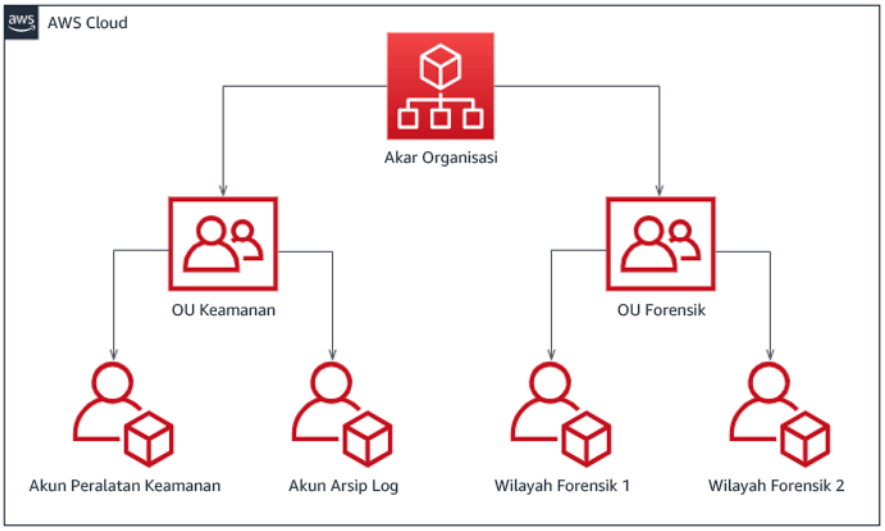
Untuk respons insiden, akan sangat membantu untuk memiliki Akun AWS struktur yang mendukung fungsi respons insiden, yang mencakup OU keamanan dan OU forensik. Dalam unit organisasi keamanan, Anda harus memiliki akun untuk:

- Arsip log: Agregat log dalam arsip log Akun AWS dengan izin terbatas.
- Alat keamanan: Memusatkan layanan keamanan dalam alat Akun AWS keamanan. Akun ini beroperasi sebagai administrator yang didelegasikan untuk layanan keamanan.

Dalam forensik unit organisasi, Anda memiliki opsi untuk menerapkan satu akun forensik atau akun-akun untuk setiap Wilayah tempat Anda beroperasi, bergantung pada mana yang paling sesuai untuk model bisnis dan operasional Anda. Jika Anda membuat akun forensik per Wilayah, Anda dapat memblokir pembuatan AWS sumber daya di luar Wilayah tersebut dan mengurangi risiko sumber daya disalin ke wilayah yang tidak diinginkan. Misalnya, jika Anda hanya beroperasi di Wilayah AS Timur (Virginia Utara) (`us-east-1`) dan AS Barat (Oregon) (`us-west-2`), maka Anda akan memiliki dua akun yang ada di forensik OU: satu untuk `us-east-1` dan satu untuk `us-west-2`.

Anda dapat membuat forensik Akun AWS untuk beberapa Wilayah. Anda harus berhati-hati dalam menyalin AWS sumber daya ke akun tersebut untuk memverifikasi bahwa Anda selaras dengan persyaratan kedaulatan data Anda. Karena penyediaan akun baru membutuhkan waktu, akun forensik harus dibuat dan digunakan jauh sebelum insiden, sehingga bisa siap digunakan oleh responden secara efektif ketika merespons insiden.

Diagram berikut menampilkan struktur akun sampel, termasuk unit organisasi forensik dengan akun forensik per Wilayah:



Struktur akun per wilayah untuk respons insiden

Menangkap cadangan dan snapshot

Menyiapkan cadangan sistem kunci dan basis data sangat penting untuk pemulihan dari insiden keamanan dan untuk tujuan forensik. Dengan memiliki cadangan, Anda dapat memulihkan sistem Anda ke keadaan aman sebelumnya. Pada AWS, Anda dapat mengambil snapshot dari berbagai sumber daya. Snapshot memberi Anda point-in-time cadangan sumber daya tersebut. Ada banyak AWS layanan yang dapat mendukung Anda dalam pencadangan dan pemulihan. Untuk membaca detail tentang layanan dan pendekatan pencadangan dan pemulihan ini, lihat Panduan Preskriptif Pencadangan dan Pemulihan dan Gunakan cadangan untuk memulihkan dari insiden keamanan.

Terutama ketika berhubungan dengan situasi seperti ransomware, sangat penting agar cadangan Anda dilindungi dengan baik. Untuk membaca panduan tentang cara mengamankan cadangan Anda, silakan lihat 10 teratas praktik terbaik untuk mengamankan cadangan di AWS . Selain mengamankan cadangan, Anda juga sebaiknya menguji proses pencadangan dan pemulihan Anda secara teratur untuk memverifikasi bahwa teknologi dan proses yang Anda miliki berfungsi sesuai harapan.

Mengotomatiskan forensik

Selama peristiwa keamanan, tim respons insiden Anda harus dapat mengumpulkan dan menganalisis bukti dengan cepat sambil mempertahankan akurasi untuk periode waktu sekitar peristiwa (seperti menangkap log yang terkait dengan peristiwa atau sumber daya tertentu atau mengumpulkan dump memori dari EC2 instans Amazon). 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, kesalahan manusia rentan terjadi

dalam pengumpulan secara manual. Untuk alasan-alasan ini, Anda harus mengembangkan dan mengimplementasikan otomatisasi untuk forensik sebisa mungkin.

AWS menawarkan sejumlah sumber daya otomatisasi untuk forensik, yang tercantum di bagian Sumber Daya berikut. Sumber daya ini adalah contoh pola forensik yang telah kami kembangkan dan telah diterapkan pelanggan. Meskipun sumber daya ini mungkin 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:

- AWS Panduan Respons Insiden Keamanan - Kembangkan Kemampuan Forensik
- AWS Panduan Respons Insiden Keamanan - Sumber Daya Forensik
- Strategi lingkungan investigasi forensik di AWS Cloud
- Cara mengotomatiskan koleksi disk forensik di AWS
- AWS Panduan Preskriptif - 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 mempersiapkan proses respons insiden Anda adalah mengembangkan playbook. Playbook respons insiden memberikan panduan preskriptif dan langkah-langkah yang harus diikuti ketika terjadi peristiwa keamanan. Struktur dan langkah yang jelas akan menyederhanakan respons dan mengurangi kemungkinan kesalahan manusia.

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

Panduan implementasi

Playbook sebaiknya dibuat untuk skenario insiden seperti:

- Insiden yang diantisipasi: Playbook harus dibuat untuk insiden yang Anda antisipasi. Hal ini termasuk ancaman seperti denial of service (DoS), ransomware, dan pembobolan kredensial.
- Temuan atau peringatan keamanan yang diketahui: Playbook harus dibuat untuk menangani temuan dan peringatan keamanan Anda yang diketahui, seperti temuan Amazon GuardDuty. Saat Anda menerima temuan GuardDuty, playbook harus memberikan langkah-langkah yang jelas untuk mencegah kesalahan penanganan atau mengabaikan peringatan. Untuk detail dan panduan remediasi selengkapnya, lihat Melakukan remediasi masalah keamanan yang ditemukan oleh GuardDuty.

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

Langkah-langkah implementasi

Item yang akan disertakan dalam playbook meliputi:

- Gambaran umum playbook: Skenario risiko atau insiden apa yang ditangani oleh playbook ini? Apa tujuan dari playbook ini?
- Persyaratan: Log, mekanisme deteksi, dan alat otomatis apa yang diperlukan untuk skenario insiden ini? Apa notifikasi yang diharapkan?
- Informasi komunikasi dan eskalasi: Siapa saja yang terlibat dan apa informasi kontak mereka? Apa saja tanggung jawab setiap pemangku kepentingan?
- Langkah respons: Di seluruh fase respons insiden, langkah taktis apa yang perlu diambil? Kueri apa yang perlu dijalankan analis? Kode apa yang perlu dijalankan untuk mencapai hasil yang diinginkan?
  - Deteksi: Bagaimana insiden tersebut akan terdeteksi?
  - Analisis: Bagaimana cakupan dampak akan ditentukan?
  - Tahan: Bagaimana insiden akan diisolasi untuk membatasi cakupan?
  - Berantas: Bagaimana ancaman akan dihilangkan dari lingkungan?
  - Pulihkan: Bagaimana sistem atau sumber daya yang terpengaruh akan dibawa kembali ke produksi?

- Hasil yang diharapkan: Setelah kueri dan kode dijalankan, apa hasil yang diharapkan dari playbook tersebut?

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.

Anti-pola umum:

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

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

Panduan implementasi

AWS menyarankan Anda untuk sebisa mungkin mengurangi atau menghilangkan kebergantungan pada kredensial berumur panjang, dan memilih kredensial sementara dan mekanisme eskalasi hak akses just-in-time. Kredensial berumur panjang rentang terkena risiko keamanan dan meningkatkan biaya overhead operasional. Untuk sebagian besar tugas manajemen, serta tugas respons insiden,

kami sarankan Anda untuk menerapkan federasi identitas bersama eskalasi 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 disetujui, pengguna akan menerima satu set kredensial AWS sementara yang dapat digunakan untuk menyelesaikan 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 yang benar untuk melakukan hal itu adalah dengan menggunakan AWS Security Token Service dan kebijakan sesi untuk mencakup 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.

Dalam kasus sebelumnya, harus ada akses kaca pecah darurat yang dikonfigurasi untuk memungkinkan penyelidikan dan perbaikan insiden dilakukan secara tepat waktu. Sebaiknya Anda menggunakan pengguna, grup, atau peran dengan izin yang sesuai untuk melakukan tugas dan mengakses sumber daya AWS. Gunakan pengguna root hanya untuk tugas yang memerlukan kredensial pengguna root. Untuk memverifikasi bahwa tim respons insiden memiliki tingkat akses yang tepat ke AWS dan sistem yang relevan lainnya, sebaiknya sediakan akun-akun khusus sejak awal. Akun-akun tersebut memerlukan akses istimewa, dan harus dikontrol dan dipantau secara ketat. Akun-akun tersebut harus dibuat dengan hak akses paling rendah 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 sebuah playbook

untuk memastikan pengguna respons insiden dibuat sebagai pengguna 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 menerapkan 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 untuk SPC AWS Organizations. 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, Anda tidak boleh membagikan akun 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 kredensial masuk dan dan token MFA pengguna root.

Untuk mengonfigurasi kebijakan IAM untuk peran respons insiden, pertimbangkan untuk menggunakan IAM Access Analyzer untuk membuat 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 respons insiden untuk mengambil peran IAM respons insiden khusus di Akun AWS lainnya. Peran-peran ini harus dikonfigurasi hanya agar dapat diambil oleh pengguna di akun keamanan, dan hubungan kepercayaan harus mewajibkan bahwa pengguna utama yang melakukan pemanggilan telah mengautentikasi dengan 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 bahwa akun IAM dan peran IAM disebutkan secara jelas agar dapat ditemukan dengan mudah di log CloudTrail. Contohnya adalah dengan menamai akun IAM dengan *<USER_ID>*-BREAK-GLASS dan peran IAM dengan BREAK-GLASS-ROLE.

CloudTrail digunakan untuk mencatat log aktivitas API di akun AWS Anda dan harus digunakan untuk mengonfigurasi peringatan tentang penggunaan peran respons insiden. Lihat postingan blog tentang konfigurasi perintanan saat kunci root digunakan. Instruksinya dapat dimodifikasi untuk mengonfigurasi metrik Amazon CloudWatch filter-to-filter pada peristiwa `AssumeRole` yang terkait dengan peran IAM 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. Ini dapat mencakup instans Amazon Elastic Compute Cloud, basis data Amazon Relational Database Service, atau platform Perangkat Lunak sebagai Layanan (SaaS). Sangat disarankan bahwa daripada menggunakan protokol asli seperti SSH atau RDP, AWS Systems Manager Session Manager digunakan untuk semua akses administratif ke instans Amazon EC2. Akses ini dapat dikontrol menggunakan IAM, yang aman dan diaudit. Dimungkinkan juga untuk mengotomatiskan bagian dari playbook Anda dengan menggunakan dokumen AWS Systems Manager Run Command, yang dapat mengurangi kesalahan pengguna dan meningkatkan waktu untuk 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, pengelolaan akun IAM respons insiden harus ditambahkan ke proses Joiner, Movers, dan Leavers Anda dan kemudian ditinjau dan diuji secara berkala untuk memverifikasi bahwa hanya akses yang dimaksudkan yang diizinkan.

Sumber daya

Dokumen terkait:

- Mengelola peningkatan akses sementara ke lingkungan AWS Anda
- Panduan Respons Insiden Keamanan AWS
- AWS Elastic Disaster Recovery
- Manajer Insiden AWS Systems 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
- Akses pecah kaca

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

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 diterapkan: 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 cakupan serta garis waktu lengkap insiden tersebut. Log juga diperlukan untuk pembuatan peringatan, yang menunjukkan terjadinya tindakan tertentu yang menarik. Sangat penting untuk memilih, mengaktifkan, menyimpan, serta mengatur mekanisme kueri dan pengambilan, serta mengatur 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-langkah implementasi

Memilih dan mengatur log untuk analisis dan 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 di sekitar sumber daya AWS bisa menjadi hal yang sulit. 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 menggunakan tag untuk mengelompokkan sumber daya berdasarkan tujuan, pemilik, lingkungan, jenis data yang diproses, dan kriteria lainnya yang Anda pilih.

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 Memberikan tag pada sumber daya AWS Anda. Anda harus terlebih dahulu menentukan tag yang ingin Anda terapkan di organisasi Anda. Setelah itu, Anda akan menerapkan dan menegakkan strategi pemberian tag ini. Untuk mendapatkan detail selengkapnya tentang implementasi dan penegakan, lihat Menerapkan strategi penandaan sumber daya AWS dengan menggunakan Kebijakan Tag AWS dan Kebijakan Kontrol Layanan (SCP).

Sumber daya

Praktik terbaik Well-Architected terkait:

- SEC04-BP01 Mengonfigurasi pencatatan log layanan dan aplikasi
- SEC04-BP02 Catat log, temuan, dan metrik di lokasi standar

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) aktor ancaman dan memungkinkan organisasi untuk melatih dan mengevaluasi kemampuan respons insiden mereka dengan merespons peristiwa siber tiruan ini yang mungkin saja akan benar-benar terjadi.

Manfaat menjalankan praktik terbaik ini: Simulasi memiliki berbagai manfaat:

- Memvalidasi kesiapan siber dan mengembangkan kepercayaan diri responden insiden Anda.
- Menguji akurasi dan efisiensi alat serta alur kerja.
- Menyempurnakan metode komunikasi dan eskalasi yang selaras dengan rencana respons insiden Anda.
- Memberikan kesempatan untuk merespons vektor yang kurang umum.

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

Panduan implementasi

Ada tiga jenis simulasi utama:

- Latihan meja: Pendekatan latihan meja dalam simulasi adalah sesi berbasis diskusi yang melibatkan berbagai pemangku kepentingan respons insiden untuk mempraktikkan peran dan tanggung jawab serta menggunakan alat komunikasi dan playbook yang telah ditetapkan. Fasilitasi latihan biasanya dapat dilakukan dalam sehari penuh di tempat virtual, bangunan fisik, atau kombinasi 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 tim responden insiden (Tim Biru) dan tim aktor ancaman simulasi (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 dalam merancang skenario yang akurat dan memungkinkan. Fokus utama dalam latihan Tim Ungu adalah pada mekanisme deteksi, alat, dan prosedur operasi standar (SOP) yang mendukung upaya respons insiden.

- Latihan Tim Merah: Dalam latihan Tim Merah, penyerang (Tim Merah) melakukan simulasi untuk mencapai tujuan tertentu atau serangkaian tujuan dari cakupan yang telah ditentukan sebelumnya. Tim pertahanan (Tim Biru) tidak harus memiliki pengetahuan tentang cakupan dan durasi latihan, sehingga memberikan penilaian yang lebih realistis tentang bagaimana mereka akan merespons insiden aktual. 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 siber secara reguler. Setiap jenis latihan dapat memberikan manfaat tersendiri bagi peserta dan organisasi secara keseluruhan, sehingga Anda dapat memilih untuk memulai dengan jenis simulasi yang kurang kompleks (seperti latihan meja) lalu beralih ke jenis simulasi yang lebih kompleks (latihan Tim Merah). Anda sebaiknya memilih jenis simulasi berdasarkan kematangan keamanan, sumber daya, dan hasil yang Anda inginkan. Beberapa pelanggan mungkin tidak memilih untuk melakukan latihan Tim Merah karena kompleksitas dan biayanya.

Langkah-langkah implementasi

Terlepas dari jenis simulasi yang Anda pilih, simulasi umumnya mengikuti langkah-langkah implementasi berikut ini:

1. Menentukan elemen latihan inti: Tentukan skenario simulasi dan tujuan simulasi. Dua hal ini harus disetujui oleh kepemimpinan.
2. Mengidentifikasi pemangku kepentingan utama: Latihan setidaknya membutuhkan fasilitator dan peserta latihan. Tergantung skenarionya, pemangku kepentingan tambahan seperti pimpinan dari departemen hukum, komunikasi, atau eksekutif dapat dilibatkan.
3. Membangun dan menguji skenario: Skenario mungkin perlu disesuaikan jika elemen tertentu tidak memungkinkan dalam pengembangannya. Tahap ini diharapkan menghasilkan skenario final.
4. Memfasilitasi simulasi: Jenis simulasi menentukan fasilitas yang digunakan (skenario tertulis atau skenario simulasi yang sangat teknis). Fasilitator harus menyelaraskan taktik fasilitasi mereka dengan objek latihan dan harus sebisa mungkin melibatkan semua peserta latihan agar hasilnya bisa optimal.

5. Mengembangkan laporan setelah tindakan (AAR): Identifikasi area yang berjalan dengan baik, area yang dapat ditingkatkan lagi, dan potensi kesenjangan. AAR harus mengukur efektivitas simulasi serta respons tim terhadap peristiwa simulasi agar kemajuan dapat dilacak dari waktu ke waktu dengan simulasi mendatang.

Sumber daya

Dokumen terkait:

- Panduan Respons Insiden AWS

Video terkait:

- AWS GameDay - Edisi Keamanan
- Menjalankan simulasi respons insiden keamanan yang efektif

Operasi

Operasi adalah hal inti dalam melakukan respons insiden. Di sinilah tindakan merespons dan meremediasi insiden keamanan terjadi. Operasi meliputi lima fase berikut: deteksi, analisis, penahanan, pemberantasan, dan pemulihan. Deskripsi fase-fase ini serta tujuannya dapat ditemukan di dalam tabel berikut.

Fase	Tujuan
Deteksi	Mengidentifikasi peristiwa keamanan potensial.
Analisis	Menentukan apakah peristiwa keamanan merupakan insiden dan menilai cakupan insiden tersebut.
Penahanan	Meminimalkan dan membatasi cakupan peristiwa keamanan.
Pemberantasan	Menghapus sumber daya atau artefak tidak sah yang terkait dengan peristiwa keamanan.

Fase	Tujuan
	Menerapkan mitigasi yang menyebabkan insiden keamanan tersebut.
Pemulihan	Mengembalikan sistem ke keadaan aman yang diketahui dan memantau sistem ini untuk memverifikasi bahwa ancaman tidak kembali.

Fase-fase ini akan berfungsi sebagai panduan ketika Anda merespons dan beroperasi pada insiden keamanan untuk merespons dengan cara yang efektif dan kuat. Tindakan aktual yang Anda ambil akan bervariasi, tergantung insiden Anda. Insiden yang melibatkan ransomware, misalnya, akan memiliki serangkaian langkah respons yang berbeda untuk diikuti dibandingkan insiden yang melibatkan bucket Amazon S3 publik. Selain itu, fase-fase ini tidak selalu terjadi secara berurutan. Setelah penahanan dan pemberantasan, Anda mungkin perlu kembali ke analisis untuk mengetahui apakah tindakan Anda efektif.

Persiapan yang menyeluruh untuk personel, proses, dan teknologi Anda adalah kunci untuk efektivitas dalam operasi. Dengan demikian, ikuti praktik terbaik yang diuraikan di bagian Persiapan agar Anda dapat secara efektif merespons peristiwa keamanan aktif.

Untuk mempelajari lebih lanjut, lihat bagian Operasi dari Panduan Respons Insiden Keamanan AWS.

Aktivitas pascainsiden

Lanskap ancaman terus berubah dan penting agar organisasi Anda memiliki kemampuan yang juga dinamis untuk melindungi lingkungan Anda secara efektif. Kunci untuk peningkatan berkelanjutan adalah melakukan iterasi pada hasil insiden dan simulasi untuk meningkatkan kemampuan Anda agar dapat secara efektif mendeteksi, merespons, dan menyelidiki kemungkinan insiden keamanan, mengurangi kemungkinan kerentanan Anda, waktu untuk merespons, dan kembali ke operasi yang aman. Mekanisme berikut dapat membantu Anda memverifikasi bahwa 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 pelajaran yang didapatkan dan kemampuan analisis akar masalah tidak hanya dapat membantu Anda meningkatkan kemampuan respons insiden, tetapi juga membantu mencegah berulang kejadian insiden. 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 diterapkan: Sedang

Panduan implementasi

Penting untuk menerapkan kerangka kerja pembelajaran dan meraih poin-poin berikut di tingkatan tinggi:

- Kapan pembelajaran diadakan?
- Apa saja yang terlibat dalam proses pembelajaran tersebut?
- Bagaimana pembelajaran dilakukan?
- Siapa yang terlibat dalam proses tersebut dan bagaimana caranya?
- Bagaimana cara mengenali area yang perlu ditingkatkan?
- Bagaimana Anda memastikan peningkatan 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-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 mendorong diskusi pembelajaran Anda:

- Apa insiden yang terjadi?
- Kapan insiden tersebut pertama kali diidentifikasi?
- Bagaimana insiden tersebut diidentifikasi?
- Sistem apa yang memunculkan peringatan tentang aktivitas tersebut?

- Sistem, layanan, dan data apa yang terlibat?
- Secara khusus, apa yang terjadi?
- Apa yang berjalan dengan baik?
- Apa yang tidak berjalan dengan baik?
- Proses atau prosedur mana yang gagal atau tidak dapat diskalakan untuk merespons insiden tersebut?
- Apa yang dapat ditingkatkan dalam bidang berikut:
  - Orang
    - Apakah orang-orang yang perlu dihubungi benar-benar tersedia dan apakah daftar kontak sudah aktual?
    - Apakah orang-orang tidak mendapatkan pelatihan atau tidak memiliki kemampuan yang diperlukan untuk merespons dan menyelidiki insiden tersebut secara efektif?
    - Apakah sumber daya yang sesuai 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 ada?
    - Apakah responden dapat memperoleh akses tepat waktu ke informasi yang diperlukan untuk merespons masalah ini?
  - Teknologi
    - Apakah sistem peringatan yang ada mampu mengidentifikasi dan memperingatkan tentang aktivitas tersebut secara efektif?
    - Bagaimana kita bisa mengurangi time-to-detection 50%?
    - Apakah peringatan yang ada perlu ditingkatkan atau apakah peringatan baru perlu dibangun 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 lebih cepat?
    - Apa yang dapat dilakukan untuk membantu mencegah (jenis) insiden ini terjadi lagi?
    - Siapa yang bertanggung jawab atas rencana peningkatan dan bagaimana cara untuk menguji apakah rencana tersebut telah diimplementasikan?
    - 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 memasukkan pembelajaran yang diambil sebagai bagian standar dari proses respons insiden, dokumentasi, dan ekspektasi di seluruh pemangku kepentingan.

Sumber daya

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS - Menetapkan kerangka kerja untuk belajar dari insiden
- NCSCCAFbimbingan - Pelajaran yang dipetik

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.

Pedoman implementasi ini berfokus pada empat bidang: organisasi dan budaya, keamanan dari pipeline, keamanan dalam 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 Lakukan peninjauan kode
- 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 tim Anda tentang praktik pengembangan dan operasi yang aman, sehingga membantu mereka membangun perangkat lunak aman dan berkualitas tinggi. Praktik ini membantu tim Anda mencegah, mendeteksi, dan memulihkan masalah keamanan lebih awal dalam siklus pengembangan. Pertimbangkan pelatihan yang mencakup pemodelan ancaman, praktik pengodean yang aman, dan penggunaan layanan untuk konfigurasi dan operasi yang aman. Beri tim Anda akses ke pelatihan melalui sumber daya mandiri, dan secara teratur kumpulkan umpan balik mereka untuk peningkatan berkelanjutan.

Hasil yang diinginkan: Anda membekali tim Anda dengan pengetahuan dan keterampilan yang diperlukan untuk merancang dan membangun perangkat lunak dengan mempertimbangkan keamanan sejak awal. Melalui pelatihan tentang pemodelan ancaman dan praktik pengembangan yang aman, tim Anda memiliki pemahaman mendalam tentang potensi risiko keamanan dan cara memitigasinya selama siklus hidup pengembangan perangkat lunak (SDLC). Pendekatan proaktif terhadap keamanan ini adalah bagian dari budaya tim Anda, dan Anda dapat mengidentifikasi serta memulihkan potensi masalah keamanan sejak dini. Hasilnya, tim Anda menghadirkan perangkat lunak dan fitur berkualitas tinggi dan aman dengan lebih efisien, sehingga mempercepat jadwal pengiriman secara keseluruhan. Anda memiliki budaya keamanan kolaboratif dan inklusif dalam organisasi Anda, di mana kepemilikan keamanan dibagi di antara semua builder.

Anti-pola umum:

- Anda menunggu sampai peninjauan keamanan, lalu mempertimbangkan karakteristik keamanan sistem.

- Anda menyerahkan semua keputusan keamanan kepada tim keamanan pusat.
- Anda tidak menyampaikan cara keputusan diambil dalam SDLC terkait ekspektasi atau kebijakan keamanan secara keseluruhan di organisasi.
- Anda terlambat melakukan 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 diterapkan: Sedang

Panduan implementasi

Untuk membangun perangkat lunak yang aman dan berkualitas tinggi, berikan pelatihan kepada tim Anda tentang praktik umum untuk pengembangan dan operasi aplikasi yang aman. Praktik ini dapat membantu tim Anda mencegah, mendeteksi, dan memulihkan masalah keamanan lebih awal dalam siklus pengembangan, sehingga dapat mempercepat jadwal pengiriman Anda.

Untuk mewujudkan praktik ini, pertimbangkan melatih tim Anda tentang pemodelan ancaman menggunakan sumber daya AWS seperti Lokakarya Pemodelan Ancaman. Pemodelan ancaman dapat membantu tim Anda memahami potensi risiko keamanan dan merancang sistem dengan mempertimbangkan keamanan sejak awal. Selain itu, Anda dapat memberikan akses ke pelatihan AWS Training and Certification, industri, atau Partner AWS tentang praktik pengembangan yang aman. Untuk detail lebih lanjut tentang pendekatan komprehensif dalam merancang, mengembangkan, mengamankan, dan beroperasi secara efisien dalam skala besar, lihat Panduan DevOps AWS.

Tentukan dan komunikasikan dengan jelas proses peninjauan keamanan organisasi Anda, dan uraikan tanggung jawab tim Anda, tim keamanan, dan pemangku kepentingan lainnya. Publikasikan panduan mandiri, contoh kode, dan templat yang menunjukkan cara memenuhi persyaratan keamanan Anda. Anda dapat menggunakan layanan AWS seperti AWS CloudFormation, AWS Cloud Development Kit (AWS CDK) (AWS CDK) Constructs, dan Service Catalog untuk menyediakan

konfigurasi yang telah disetujui sebelumnya dan aman serta mengurangi kebutuhan untuk pengaturan kustom.

Dapatkan umpan balik secara rutin dari tim Anda terkait pengalaman mereka seputar pelatihan dan proses peninjauan keamanan, dan gunakan umpan balik tersebut untuk terus melakukan peningkatan. Lakukan game day atau aktivitas bug bash untuk mengidentifikasi dan mengatasi masalah keamanan sambil meningkatkan keterampilan tim Anda.

Langkah-langkah implementasi

1. Identifikasi kebutuhan pelatihan: Evaluasi tingkat keterampilan saat ini dan kesenjangan pengetahuan dalam tim Anda mengenai praktik pengembangan yang aman melalui survei, peninjauan kode, atau diskusi dengan anggota tim.
2. Rencanakan pelatihan: Berdasarkan kebutuhan yang teridentifikasi, buat rencana pelatihan yang mencakup topik yang relevan seperti pemodelan ancaman, praktik pengodean aman, pengujian keamanan, dan praktik deployment yang aman. Gunakan sumber daya seperti Lokakarya Pemodelan Ancaman, AWS Training and Certification, dan program pelatihan industri atau Partner AWS.
3. Jadwalkan dan berikan pelatihan: Jadwalkan sesi pelatihan atau lokakarya reguler untuk tim Anda. Hal ini dapat dipimpin oleh instruktur atau mandiri, tergantung pada preferensi dan ketersediaan tim Anda. Anjurkan latihan praktik langsung dan sediakan contoh praktis untuk memperkuat pembelajaran.
4. Tentukan proses peninjauan keamanan: Bekerjasamalah dengan tim keamanan Anda dan pemangku kepentingan lainnya untuk menentukan secara jelas proses peninjauan keamanan untuk aplikasi Anda. Dokumentasikan tanggung jawab setiap tim atau individu yang terlibat dalam proses tersebut, termasuk tim pengembangan Anda, tim keamanan, dan pemangku kepentingan terkait lainnya.
5. Buat sumber daya mandiri: Kembangkan panduan mandiri, contoh kode, dan templat yang menunjukkan cara memenuhi persyaratan keamanan organisasi Anda. Pertimbangkan layanan AWS seperti CloudFormation, AWS CDK Constructs, dan Service Catalog untuk menyediakan konfigurasi yang telah disetujui sebelumnya dan aman serta mengurangi kebutuhan untuk pengaturan kustom.
6. Lakukan komunikasi dan sosialisasi: Komunikasikan secara efektif proses peninjauan keamanan dan sumber daya mandiri yang tersedia kepada tim Anda. Lakukan sesi pelatihan atau lokakarya untuk membiasakan mereka dengan sumber daya ini, dan verifikasi bahwa mereka memahami cara menggunakannya.

7. Kumpulkan umpan balik dan lakukan peningkatan: Dapatkan umpan balik secara rutin dari tim Anda terkait pengalaman mereka seputar pelatihan dan proses peninjauan keamanan. Gunakan umpan balik ini untuk mengidentifikasi area yang memerlukan peningkatan serta terus menyempurnakan materi pelatihan, sumber daya mandiri, dan proses peninjauan keamanan.
8. Lakukan latihan keamanan: Adakan game day atau aktivitas bug bash untuk mengidentifikasi dan mengatasi masalah keamanan dalam aplikasi Anda. Latihan ini tidak hanya membantu mengungkap potensi kerentanan, tetapi juga berfungsi sebagai kesempatan pembelajaran praktis bagi tim Anda yang meningkatkan keterampilan mereka dalam pengembangan dan operasi yang aman.
9. Terus belajar dan lakukan peningkatan: Dorong tim Anda untuk terus mengikuti informasi tentang praktik, alat, dan teknik pengembangan terbaru yang aman. Tinjau dan perbarui materi dan sumber daya pelatihan Anda secara berkala untuk mencerminkan lanskap dan praktik terbaik keamanan yang terus berubah.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC11-BP08 Buat program yang menanamkan kepemilikan keamanan dalam tim beban kerja

Dokumen terkait:

- AWS Training dan Sertifikasi
- Bagaimana cara berpikir tentang tata kelola keamanan cloud
- Cara melakukan pendekatan terhadap pemodelan ancaman
- Pelatihan akselerasi – AWS Skills Guild
- AWS DevOps Sagas

Video terkait:

- Keamanan proaktif: Pertimbangan dan Pendekatan

Contoh terkait:

- Lokakarya pemodelan ancaman

- Kesadaran industri untuk para developer

Layanan terkait:

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK) (AWS CDK) Constructs
- Katalog Layanan

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, sehingga mengurangi risiko masalah keamanan dalam perangkat lunak yang disediakan.

Hasil yang diinginkan: Tujuan dari pengujian otomatis adalah untuk menyediakan cara terprogram untuk mendeteksi potensi masalah lebih awal dan seringkali 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.

Anti-pola 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 diterapkan: 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 Source Code Security Analyzers, yang mencakup alat-alat sumber terbuka untuk Byte Code Scanners dan Binary Code Scanners.

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 Q Developer menyediakan rekomendasi kode, termasuk pemindaian keamanan, di IDE yang dimiliki builder. Keamanan Amazon CodeGuru dapat mengidentifikasi masalah kritis, masalah keamanan, dan bug yang sulit ditemukan selama pengembangan aplikasi, dan memberikan rekomendasi untuk meningkatkan kualitas kode. Mengekstrak Software Bill of Material (SBOM) juga memungkinkan Anda mengekstrak data formal yang berisi detail dan keterkaitan berbagai komponen yang digunakan dalam membuat perangkat lunak Anda. Hal ini memungkinkan Anda menentukan manajemen kerentanan, dan dengan cepat mengidentifikasi dependensi perangkat lunak atau komponen dan risiko rantai pasokan.

Lokakarya Keamanan untuk Developer menggunakan alat-alat developer AWS, seperti AWS CodeBuild, AWS CodeCommit, dan AWS CodePipeline, untuk melakukan otomatisasi terhadap pipeline perilisan yang mencakup 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. Peninjauan ini membuat postur keamanan aplikasi yang efektif, dan memberikan umpan balik yang dapat ditindaklanjuti kepada builder untuk menangani potensi masalah.

Langkah-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.
- Automated Security Helper (ASH) adalah contoh alat pemindaian keamanan kode sumber terbuka.
- Melakukan pengujian atau analisis kode menggunakan alat-alat otomatis, seperti Amazon Q Developer yang terintegrasi dengan IDE developer, dan Keamanan Amazon CodeGuru yang digunakan untuk memindai kode pada saat melakukan commit, akan membantu builder mendapatkan umpan balik pada waktu yang tepat.
- Saat membangun menggunakan AWS Lambda, Anda dapat menggunakan Amazon Inspector untuk memindai kode aplikasi yang ada dalam fungsi Anda.
- 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.
- Untuk mengekstrak SBOM sebagai bagian dari pipeline CI/CD Anda, gunakan Amazon Inspector SBOM Generator untuk menghasilkan SBOM untuk arsip, image kontainer, direktori, sistem lokal, dan binari Go dan Rust yang dikompilasi dalam format SBOM CycloneDX.

Sumber daya

Praktik-praktik terbaik terkait:

- Panduan DevOps: DL.CR.3 Tetapkan kriteria penyelesaian yang jelas untuk tugas kode

Dokumen terkait:

- Pengiriman Berkelanjutan dan Deployment Berkelanjutan

- Mitra Kompetensi DevOps AWS
- Mitra Kompetensi Keamanan AWS untuk Keamanan Aplikasi
- Memilih pendekatan CI/CD Well-Architected
- Deteksi Rahasia di Keamanan Amazon CodeGuru
- Pustaka Deteksi Keamanan Amazon CodeGuru
- Mempercepat deployment di AWS dengan tata kelola yang efektif
- Bagaimana pendekatan AWS dalam melakukan otomatisasi deployment secara aman dan otonom
- Bagaimana Keamanan Amazon CodeGuru membantu Anda menyeimbangkan keamanan dan kecepatan secara efektif

Video terkait:

- Hands-off: Mengotomatiskan pipeline pengiriman berkelanjutan di Amazon
- Mengotomatiskan pipeline CI/CD lintas akun
- Proses Pengembangan Perangkat Lunak di Amazon
- Menguji perangkat lunak dan sistem di Amazon

Contoh terkait:

- Kesadaran industri untuk para developer
- Automated Security Helper (ASH)
- Tata Kelola AWS CodePipeline - GitHub

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-kontrol detektif 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: Pengujian penetrasi digunakan untuk mendeteksi, memulihkan, dan melakukan validasi terhadap properti 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.

Anti-pola 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, sehingga 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 diterapkan: 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-langkah implementasi

- Memiliki proses terstruktur untuk bagaimana pengujian penetrasi dicakup, mendasarkan proses ini pada model ancaman adalah sebuah cara yang baik untuk 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-praktik terbaik terkait:

- SEC11-BP01 Pelatihan untuk keamanan aplikasi
- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Uji Penetrasi AWS memberikan panduan rinci untuk melakukan pengujian penetrasi pada AWS
- Mempercepat deployment di AWS dengan tata kelola yang efektif
- Mitra Kompetensi Keamanan AWS
- Lakukan modernisasi terhadap arsitektur uji penetrasi Anda di AWS Fargate
- Simulator Injeksi Kesalahan AWS

Contoh terkait:

- Mengotomatiskan pengujian API dengan AWS CodePipeline (GitHub)
- Pembantu keamanan otomatis (GitHub)

SEC11-BP04 Lakukan peninjauan kode

Terapkan peninjauan kode untuk membantu memverifikasi kualitas dan keamanan perangkat lunak yang sedang dikembangkan. Peninjauan kode mengharuskan anggota tim selain penulis kode asli meninjau kode untuk menemukan potensi masalah, kerentanan, dan kepatuhan terhadap standar dan praktik terbaik pengodean. Proses ini membantu menemukan kesalahan, inkonsistensi, dan kelemahan keamanan yang mungkin terlewat oleh developer asli. Gunakan alat otomatis untuk membantu peninjauan kode.

Hasil yang diinginkan: Anda menyertakan peninjauan kode selama pengembangan untuk meningkatkan kualitas perangkat lunak yang sedang ditulis. Anda meningkatkan keterampilan anggota tim yang kurang berpengalaman melalui pembelajaran yang diidentifikasi selama peninjauan kode. Anda mengidentifikasi peluang untuk otomatisasi dan mendukung proses peninjauan kode menggunakan alat dan pengujian otomatis.

Anti-pola umum:

- Anda tidak meninjau kode sebelum deployment.
- Orang yang sama menulis dan meninjau kode.

- Anda tidak menggunakan otomatisasi untuk membantu atau melakukan orkestrasi peninjauan kode.
- Anda 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 diterapkan: Sedang

Panduan implementasi

Peninjauan kode membantu memverifikasi kualitas dan keamanan perangkat lunak selama pengembangan. Peninjauan manual mengharuskan anggota tim selain penulis kode asli meninjau kode untuk menemukan potensi masalah, kerentanan, dan kepatuhan terhadap standar dan praktik terbaik pengodean. Proses ini membantu menemukan kesalahan, inkonsistensi, dan kelemahan keamanan yang mungkin terlewat oleh developer asli.

Pertimbangkan Keamanan Amazon CodeGuru untuk membantu melakukan peninjauan kode otomatis. Keamanan CodeGuru menggunakan machine learning dan penalaran otomatis untuk menganalisis kode Anda serta mengidentifikasi potensi kerentanan keamanan dan masalah pengodean. Integrasikan peninjauan kode otomatis dengan repositori kode yang ada serta pipeline integrasi berkelanjutan dan deployment berkelanjutan (CI/CD).

Langkah-langkah implementasi

1. Buat proses peninjauan kode:
   - Tentukan kapan peninjauan kode harus dilakukan, seperti sebelum menggabungkan kode ke cabang utama atau sebelum melakukan deployment ke produksi.
   - Tentukan siapa yang harus terlibat dalam proses peninjauan kode, seperti anggota tim, developer senior, dan ahli keamanan.
   - Tentukan metodologi peninjauan kode, termasuk proses dan alat yang akan digunakan.
2. Siapkan alat peninjauan kode:

- Evaluasi dan pilih alat peninjauan kode yang sesuai dengan kebutuhan tim Anda, seperti GitHub Pull Requests atau Keamanan CodeGuru
- Integrasikan alat yang dipilih dengan repositori kode yang ada dan pipeline CI/CD Anda.
- Konfigurasikan alat untuk memberlakukan persyaratan peninjauan kode, seperti jumlah minimum peninjau dan aturan persetujuan.

3. Tentukan daftar periksa dan pedoman peninjauan kode:
   - Buat daftar periksa atau pedoman peninjauan kode yang menguraikan apa yang harus ditinjau. Pertimbangkan faktor-faktor seperti kualitas kode, kerentanan keamanan, kepatuhan terhadap standar pengodean, dan kinerja.
   - Bagikan daftar periksa atau pedoman kepada tim pengembangan, dan pastikan bahwa semua orang memahami ekspektasinya.
4. Latih developer tentang praktik terbaik peninjauan kode:
   - Berikan pelatihan kepada tim Anda tentang cara melakukan peninjauan kode yang efektif.
   - Edukasi tim Anda tentang prinsip keamanan aplikasi dan kerentanan umum yang harus dicari selama peninjauan.
   - Dorong praktik berbagi pengetahuan dan sesi pemrograman berpasangan untuk meningkatkan keterampilan anggota tim yang kurang berpengalaman.
5. Terapkan proses peninjauan kode:
   - Integrasikan langkah peninjauan kode ke dalam alur kerja pengembangan Anda, seperti membuat permintaan pull dan menetapkan peninjau.
   - Wajibkan agar perubahan kode menjalani peninjauan kode sebelum penggabungan atau deployment.
   - Dorong komunikasi terbuka dan umpan balik yang konstruktif selama proses peninjauan.
6. Pantau dan tingkatkan:
   - Tinjau secara teratur efektivitas proses peninjauan kode Anda dan kumpulkan umpan balik dari tim.
   - Identifikasi peluang untuk otomatisasi atau peningkatan alat guna menyederhanakan proses peninjauan kode.
   - Terus perbarui dan efektifkan daftar periksa atau pedoman peninjauan kode berdasarkan pembelajaran dan praktik terbaik industri.
7. Tumbuhkan budaya peninjauan kode:
   - Tekankan pentingnya peninjauan kode untuk menjaga kualitas dan keamanan kode.

- Rayakan keberhasilan dan pembelajaran dari proses peninjauan kode.
- Dorong lingkungan kolaboratif dan suportif di mana developer merasa nyaman dalam memberikan dan menerima umpan balik.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Panduan DevOps: DL.CR.2 Lakukan proses penilaian sejawat untuk perubahan kode
- Tentang permintaan pull di GitHub

Contoh terkait:

- Lakukan otomatisasi peninjauan kode dengan Keamanan Amazon CodeGuru
- Mengotomatiskan deteksi kerentanan keamanan dan bug di pipeline CI/CD dengan menggunakan CLI Keamanan Amazon CodeGuru

Video terkait:

- Peningkatan kualitas kode yang berkelanjutan dengan Keamanan Amazon CodeGuru

SEC11-BP05 Pusatkan layanan untuk paket dan dependensi

Berikan layanan terpusat agar tim Anda dapat memperoleh paket perangkat lunak dan dependensi lainnya. Hal ini akan 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: Anda membangun beban kerja Anda dari paket perangkat lunak eksternal selain kode yang Anda tulis. Hal ini akan mempermudah Anda untuk mengimplementasikan fungsionalitas yang berulang kali digunakan, seperti pengurai JSON atau pustaka enkripsi. Anda memusatkan sumber untuk paket dan dependensi ini sehingga tim keamanan Anda dapat

memvalidasinya sebelum digunakan. Anda menggunakan pendekatan ini sehubungan dengan alur pengujian manual dan otomatis untuk meningkatkan keyakinan terhadap kualitas perangkat lunak yang sedang Anda kembangkan.

Anti-pola umum:

- Anda menarik paket dari repositori arbitrer di internet.
- Anda 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 sedang dibangun.
- Dapat memberi tahu tim beban kerja saat paket perlu diperbarui berdasarkan pemahaman siapa yang menggunakan apa.
- Menurunkan risiko terjadinya penyertaan paket bermasalah di perangkat lunak Anda.

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

Panduan implementasi

Sediakan layanan tersentralisasi untuk paket dan dependensi dengan cara yang mudah digunakan bagi builder. Layanan tersentralisasi dapat dipusatkan secara logis daripada diimplementasikan sebagai sistem monolitik. Pendekatan ini memungkinkan Anda menyediakan layanan dengan cara yang memenuhi kebutuhan builder Anda. Anda harus menerapkan cara-cara yang efisien untuk menambahkan paket ke repositori ketika ada pembaruan yang terjadi atau ada persyaratan baru yang muncul. Layanan-layanan AWS seperti AWS CodeArtifact atau solusi partner AWS yang serupa menyediakan cara untuk memberikan kemampuan ini.

Langkah-langkah implementasi

- Implementasikan layanan repositori tersentralisasi 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 sebuah 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 adanya potensi dampak masalah yang baru ditemukan.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Panduan DevOps: DL.CS.2 Tanda tangani artefak kode setelah setiap build
- Tingkat rantai pasokan untuk Artefak Perangkat Lunak (SLSA)

Contoh terkait:

- Mempercepat deployment di AWS dengan tata kelola yang efektif
- Perkuat keamanan paket Anda menggunakan toolkit CodeArtifact Package Origin Control
- Pipeline Penerbitan Paket Multi-Wilayah (GitHub)
- Menerbitkan Modul Node.js di AWS CodeArtifact dengan menggunakan AWS CodePipeline (GitHub)
- Contoh Pipeline CodeArtifact Java AWS CDK (GitHub)
- Distribusikan paket.NET NuGet privat dengan AWS CodeArtifact (GitHub)

Video terkait:

- Keamanan proaktif: Pertimbangan dan Pendekatan
- Filosofi Keamanan AWS (re: Invent 2017)
- Ketika keamanan, keselamatan, dan urgensi semuanya penting: Menangani Log4Shell

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: Versi beban kerja yang Anda uji adalah versi yang Anda deploy, dan deployment dilakukan secara konsisten setiap saat. Anda mengeksternalisasi konfigurasi beban kerja Anda, sehingga membantu Anda melakukan deployment ke berbagai lingkungan tanpa perubahan. Anda menerapkan penandatanganan kriptografis terhadap paket perangkat lunak untuk memastikan bahwa tidak ada yang berubah di antara lingkungan.

Anti-pola 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.
- Kemampuan 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 diterapkan: Tinggi

Panduan implementasi

Untuk mempertahankan infrastruktur aplikasi yang efektif dan andal, terapkan praktik deployment yang aman dan otomatis. Praktik ini mencakup penghapusan akses manusia yang persisten dari lingkungan produksi, dengan menggunakan alat CI/CD untuk deployment, dan mengeksternalisasi

data konfigurasi khusus lingkungan. Dengan mengikuti pendekatan ini, Anda dapat meningkatkan keamanan, mengurangi risiko kesalahan manusia, dan menyederhanakan proses deployment.

Anda dapat membangun struktur Akun AWS Anda untuk menghapus akses manusia yang persisten dari lingkungan produksi. Praktik ini meminimalkan risiko perubahan yang tidak sah atau modifikasi yang tidak disengaja, sehingga meningkatkan integritas sistem produksi Anda. Alih-alih akses manusia langsung, Anda dapat menggunakan alat CI/CD seperti AWS CodeBuild dan AWS CodePipeline untuk melakukan deployment. Anda dapat menggunakan layanan ini untuk mengotomatiskan proses pembuatan, pengujian, dan deployment, sehingga mengurangi intervensi manual dan meningkatkan konsistensi.

Untuk lebih meningkatkan keamanan dan keterlacakan, Anda dapat menandatangani paket aplikasi setelah diuji dan memvalidasi tanda tangan ini selama deployment. Untuk melakukannya, gunakan alat kriptografi seperti AWS Signer atau AWS Key Management Service (AWS KMS). Dengan menandatangani dan memverifikasi paket, Anda dapat memastikan bahwa Anda hanya melakukan deployment kode yang telah diotorisasi dan divalidasi ke lingkungan Anda.

Selain itu, tim Anda dapat merancang beban kerja Anda untuk mendapatkan data konfigurasi khusus lingkungan dari sumber eksternal, seperti Penyimpanan Parameter AWS Systems Manager. Praktik ini memisahkan kode aplikasi dari data konfigurasi, sehingga membantu Anda mengelola dan memperbarui konfigurasi secara independen tanpa memodifikasi kode aplikasi itu sendiri.

Untuk menyederhanakan penyediaan dan manajemen infrastruktur, pertimbangkan menggunakan alat infrastruktur sebagai kode (IaC) seperti AWS CloudFormation atau AWS CDK. Anda dapat menggunakan alat ini untuk mendefinisikan infrastruktur sebagai kode Anda, sehingga meningkatkan konsistensi dan kemampuan pengulangan deployment di berbagai lingkungan.

Pertimbangkan deployment canary untuk memvalidasi keberhasilan deployment perangkat lunak Anda. Deployment canary mencakup peluncuran perubahan ke subset instans atau pengguna sebelum melakukan deployment ke seluruh lingkungan produksi. Anda kemudian dapat memantau dampak perubahan dan melakukan rollback jika perlu, sehingga meminimalkan risiko masalah yang meluas.

Ikuti rekomendasi yang diuraikan dalam laporan resmi Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun. Laporan resmi ini memberikan panduan tentang memisahkan lingkungan (seperti pengembangan, staging, dan produksi) ke dalam Akun AWS yang berbeda, sehingga lebih meningkatkan keamanan dan isolasi.

Langkah-langkah implementasi

1. Siapkan struktur Akun AWS:
   - Ikuti panduan dalam laporan resmi Menyusun Lingkungan AWS Anda Menggunakan Beberapa Akun guna membuat Akun AWS terpisah untuk lingkungan yang berbeda (misalnya, pengembangan, staging, dan produksi).
   - Konfigurasikan kontrol akses dan izin yang sesuai untuk setiap akun guna membatasi akses manusia langsung ke lingkungan produksi.
2. Terapkan pipeline CI/CD:
   - Siapkan pipeline CI/CD menggunakan layanan seperti AWS CodeBuild dan AWS CodePipeline.
   - Konfigurasikan pipeline untuk membangun, menguji, dan melakukan deployment kode aplikasi Anda secara otomatis ke lingkungan masing-masing.
   - Integrasikan repositori kode dengan pipeline CI/CD untuk kontrol versi dan manajemen kode.
3. Tanda tangani dan verifikasi paket aplikasi:
   - Gunakan AWS Signer atau AWS Key Management Service (AWS KMS) untuk menandatangani paket aplikasi Anda setelah diuji dan divalidasi.
   - Konfigurasikan proses deployment untuk memverifikasi tanda tangan paket aplikasi sebelum Anda melakukan deployment-nya ke lingkungan target.
4. Eksternalisasi data konfigurasi:
   - Simpan data konfigurasi khusus lingkungan di Penyimpanan Parameter AWS Systems Manager.
   - Ubah kode aplikasi Anda untuk mengambil data konfigurasi dari Penyimpanan Parameter selama deployment atau runtime.
5. Terapkan infrastruktur sebagai kode (IaC):
   - Gunakan alat IaC seperti AWS CloudFormation atau AWS CDK untuk mendefinisikan dan mengelola infrastruktur sebagai kode Anda.
   - Buat templat CloudFormation atau skrip CDK untuk menyediakan dan mengonfigurasi sumber daya AWS yang diperlukan untuk aplikasi Anda.
   - Integrasikan IaC dengan pipeline CI/CD Anda untuk melakukan deployment perubahan infrastruktur secara otomatis bersamaan dengan perubahan kode aplikasi.
6. Terapkan deployment canary:

- Konfigurasikan proses deployment Anda untuk mendukung deployment canary, di mana perubahan diluncurkan ke subset instans atau pengguna sebelum Anda melakukan deployment-nya ke seluruh lingkungan produksi.
- Gunakan layanan seperti AWS CodeDeploy atau AWS ECS untuk mengelola deployment canary dan memantau dampak perubahan.
- Terapkan mekanisme rollback untuk kembali ke versi stabil sebelumnya jika masalah terdeteksi selama deployment canary.

7. Pantau dan audit:
   - Siapkan mekanisme pemantauan dan pencatatan log untuk melacak deployment, kinerja aplikasi, dan perubahan infrastruktur.
   - Gunakan layanan seperti Amazon CloudWatch dan AWS CloudTrail untuk mengumpulkan serta menganalisis log dan metrik.
   - Terapkan pemeriksaan audit dan kepatuhan untuk memverifikasi kepatuhan terhadap praktik terbaik keamanan dan persyaratan peraturan.
8. Terus tingkatkan:
   - Tinjau dan perbarui praktik deployment Anda secara berkala, serta terapkan umpan balik dan pelajaran yang diperoleh dari deployment sebelumnya.
   - Otomatiskan sebanyak mungkin proses deployment untuk mengurangi intervensi manual dan potensi kesalahan manusia.
   - Berkolaborasilah dengan tim lintas fungsi (misalnya, operasi atau keamanan) untuk menyelaraskan dan terus meningkatkan praktik deployment.

Dengan mengikuti langkah-langkah ini, Anda dapat menerapkan praktik deployment yang aman dan otomatis di lingkungan AWS Anda, sehingga meningkatkan keamanan, mengurangi risiko kesalahan manusia, dan menyederhanakan proses deployment.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis
- DL.CI.2 Picu proses build secara otomatis berdasarkan modifikasi kode sumber

Dokumen terkait:

- Mempercepat deployment di AWS dengan tata kelola yang efektif
- Melakukan otomatisasi deployment secara aman dan otonom
- Penandatanganan kode dengan menggunakan AWS Certificate Manager Private CA dan kunci asimetris AWS Key Management Service
- Penandatanganan Kode, Kontrol Integritas dan Kepercayaan untuk AWS Lambda

Video terkait:

- Hands-off: Mengotomatiskan pipeline pengiriman berkelanjutan di Amazon

Contoh terkait:

- Deployment Blue/Green dengan AWS Fargate

SEC11-BP07 Nilai karakteristik keamanan pipeline secara teratur

Terapkan prinsip-prinsip Pilar Keamanan Well-Architected pada pipeline Anda, dengan perhatian khusus pada pemisahan izin. Nilai karakteristik keamanan infrastruktur pipeline Anda secara teratur. Mengelola keamanan dari pipeline secara efektif akan memungkinkan Anda memberikan keamanan perangkat lunak yang lolos melalui pipeline.

Hasil yang diinginkan: Pipeline yang Anda gunakan untuk melakukan pengembangan dan deployment perangkat lunak Anda mengikuti praktik yang direkomendasikan yang sama seperti beban kerja lainnya di lingkungan Anda. Pengujian yang Anda terapkan di pipeline Anda tidak dapat diedit oleh tim yang menggunakannya. Anda hanya memberikan izin yang diperlukan ke pipeline untuk deployment yang dilakukan menggunakan kredensial sementara. Anda menerapkan pengamanan untuk mencegah pipeline di-deploy ke lingkungan yang salah. Anda mengonfigurasi pipeline untuk memancarkan status sehingga integritas lingkungan build dapat divalidasi.

Anti-pola 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 diterapkan: Tinggi

Panduan implementasi

Pipeline deployment Anda adalah komponen penting dari siklus hidup pengembangan perangkat lunak Anda serta harus mengikuti prinsip dan praktik keamanan yang sama seperti beban kerja lainnya di lingkungan Anda. Hal ini termasuk menerapkan kontrol akses yang tepat, memvalidasi input, serta secara teratur meninjau dan mengaudit izin yang terkait dengan infrastruktur CI/CD Anda.

Verifikasi bahwa tim yang bertanggung jawab untuk membangun dan menerapkan aplikasi tidak memiliki kemampuan untuk mengedit atau memintas pengujian keamanan dan pemeriksaan yang diterapkan di pipeline Anda. Pemisahan perhatian ini membantu menjaga integritas proses build dan deployment Anda.

Sebagai titik awal, pertimbangkan menggunakan Arsitektur Referensi Pipelines Deployment AWS. Arsitektur referensi ini menyediakan fondasi yang aman dan dapat diskalakan untuk membangun pipeline CI/CD Anda di AWS.

Selain itu, Anda dapat menggunakan layanan seperti AWS Identity and Access Management Access Analyzer untuk membuat kebijakan IAM dengan hak akses paling rendah untuk izin pipeline dan sebagai langkah dalam pipeline guna memverifikasi izin beban kerja. Hal ini membantu memverifikasi bahwa pipeline dan beban kerja Anda hanya memiliki izin yang diperlukan untuk fungsi spesifiknya, sehingga mengurangi risiko akses atau tindakan yang tidak sah.

Langkah-langkah implementasi

- Mulailah dengan Arsitektur Referensi Pipeline Deployment AWS.
- Pertimbangkan untuk menggunakan AWS IAM Access Analyzer untuk secara terprogram membuat kebijakan IAM dengan hak akses paling rendah untuk pipeline tersebut.
- Integrasikan pipeline Anda dengan pemantauan dan peringatan sehingga Anda akan mendapatkan notifikasi tentang aktivitas yang tidak terduga atau tidak normal, untuk layanan-layanan terkelola

AWS, Amazon EventBridge akan memungkinkan Anda untuk merutekan data ke target seperti AWS Lambda atau Amazon Simple Notification Service (Amazon SNS).

Sumber daya

Dokumen terkait:

- Arsitektur Referensi Pipeline Deployment AWS
- Memantau 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: Anda telah menanamkan kepemilikan dan pengambilan keputusan keamanan dalam tim Anda. Anda telah melatih tim Anda tentang cara memikirkan keamanan atau telah menambah tim mereka dengan orang-orang keamanan yang tertanam atau terkait. Tim Anda membuat keputusan keamanan berkualitas lebih tinggi di awal siklus pengembangan sebagai hasilnya.

Anti-pola 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 sedang ditulis.
- Peluang untuk mengidentifikasi dan memahami masalah-masalah sistemik atau area peningkatan bernilai tinggi.
- Penurunan jumlah pengerjaan ulang yang diperlukan akibat adanya temuan dari peninjauan keamanan.
- Peningkatan persepsi fungsi keamanan.

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

Panduan implementasi

Mulailah 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 yang akan digunakan, 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 ini, penting untuk memilih metrik yang dapat digunakan untuk menunjukkan nilai program. Belajar dari cara AWS menangani masalah ini akan menjadi 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-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-metrik ini untuk membuat peningkatan.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC11-BP01 Pelatihan untuk keamanan aplikasi
- SEC11-BP02 Otomatiskan pengujian sepanjang siklus hidup pengembangan dan rilis

Dokumen terkait:

- Cara melakukan pendekatan terhadap pemodelan ancaman
- Cara memikirkan tata kelola keamanan cloud
- Cara AWS membangun program Security Guardians, mekanisme untuk mendistribusikan kepemilikan keamanan
- Cara membangun program Security Guardians untuk mendistribusikan kepemilikan keamanan

Video terkait:

- Keamanan proaktif: Pertimbangan dan pendekatan
- Kiat alat dan budaya AppSec dari AWS dan Toyota Motor Amerika Utara

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 dan AWS fitur dan layanan program yang dibahas dalam paper ini.

AWS berusaha untuk membantu Anda membangun dan mengoperasikan arsitektur yang melindungi informasi, sistem, dan aset sambil memberikan nilai bisnis.

Kontributor

Individu dan organisasi berikut berkontribusi terhadap dokumen ini:

- Jay Michael, Arsitek Solusi Pimpinan Keamanan Utama, Amazon Web Services
- Kiaan Sumeet, Konsultan Keamanan Utama, Amazon Web Services
- Michael Fischer, Arsitek Solusi Utama, Amazon Web Services
- Conor Colgan, Arsitek Solusi Utama, Amazon Web Services
- Dave Walker, Arsitek Solusi Utama, Keamanan & Kepatuhan, Amazon Web Services
- Patrick Palmer, Arsitek Solusi Utama, Keamanan & Kepatuhan, Amazon Web Services
- Monka Vu Minh, Security Consultant, Amazon Web Services
- Kurt Kumar, Security Consultant, Amazon Web Services
- Fahima Khan, Security Solutions Architect, Amazon Web Services
- Mutaz Hajeer, Senior Security Solutions Architect, Amazon Web Services
- Luis Pastor, Senior Security Solutions Architect, Amazon Web Services
- Colin Igbokwe, Senior Security Solutions Architect, Amazon Web Services
- Geoff Sweet, Senior Security Solutions Architect, Amazon Web Services
- Anthony Harvey, Senior Security Solutions Architect, Amazon Web Services
- Sowjanya Rajavaram, Senior Security Solutions Architect, Amazon Web Services
- Krishna Prasad, Senior Solutions Architect, Amazon Web Services
- Faisal Farooq, Senior Solutions Architect, Amazon Web Services
- Arun Krishnaswamy, Senior Solutions Architect, Amazon Web Services
- Dan Girard, Senior Solutions Architect, Amazon Web Services
- Marc Luescher, Senior Solutions Architect, Amazon Web Services
- Kyle Nicodemus, Senior Technical Account Manager, Amazon Web Services
- Irina Szabo, Senior Technical Account Manager, Amazon Web Services
- Arun Sivaraman, Solutions Architect, Amazon Web Services
- Stephen Novak, Technical Account Manager, Amazon Web Services
- Jonathan Risbrook, Technical Account Manager, Amazon Web Services
- Freddy Kasprzykowski, Practice Manager - Global Financial Services, Amazon Web Services
- Pat Gaw, Konsultan Keamanan Utama, Amazon Web Services

- Jason Garman, Arsitek Solusi Keamanan Utama, Amazon Web Services
- Mark Keating, Arsitek Solusi Keamanan Utama, Amazon Web Services
- Zach Miller, Arsitek Solusi Keamanan Utama, Amazon Web Services
- Maitreya Ranganath, Arsitek Solusi Keamanan Utama, Amazon Web Services
- Reef Dsouza, Arsitek Solusi Utama, Amazon Web Services
- Brad Burnett, Security Solutions Architect, Amazon Web Services
- Matt Saner, Senior Manager, Security Solutions Architecture, Amazon Web Services
- Priyank Ghedia, Senior Security Solutions Architect, Amazon Web Services
- Arthur Mnev, Senior Security Solutions Architect, Amazon Web Services
- Kyle Dickinson, Senior Security Solutions Architect, Amazon Web Services
- Kevin Boland, Senior Security Solutions Architect, Amazon Web Services
- Anna McAbee, Senior Security Solutions Architect, Amazon Web Services
- Recep Meric Degirmenci, Senior Security Solutions Architect, Amazon Web Services
- Daniel Salzedo, Senior Security Technical Product Manager, Amazon Web Services
- Jake Izumi, Senior Solutions Architect, Amazon Web Services
- Bert Bullough, Senior Solutions Architect, Amazon Web Services
- Robert McCall, Solutions Architect, Amazon Web Services
- Angela Chao, ESL TAM, Dukungan Perusahaan AWS, Amazon Web Services
- Pratima Singh, Spesifikasi Keamanan Senior ANZ. Solutions Architect, Amazon Web Services
- Darran Boyd, Principal, Office of the CISO, AWS Security, Amazon Web Services
- Byron Pogson, Senior Security Solutions Architect, Amazon Web Services

Sumber bacaan lebih lanjut

Untuk mendapatkan bantuan tambahan, silakan konsultasikan dengan sumber berikut:

- Laporan resmi Kerangka Kerja AWS Well-Architected
- Pusat Arsitektur AWS

Revisi dokumen

Untuk mengetahui jika ada perubahan pada laporan resmi ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Panduan praktik terbaik yang sudah diperbarui	Praktik terbaik diperbarui dengan panduan baru di area berikut: SEC 2, SEC 3, SEC 4, SEC 6, SEC 7, SEC 8, SEC 9, SEC 10, dan SEC 11. Panduan telah diperbarui dan disempurnakan untuk seluruh pilar.	6 November 2024
Panduan praktik terbaik yang sudah diperbarui	Pembaruan praktik terbaik skala besar sudah dilakukan di seluruh pilar. Beberapa praktik terbaik disusun ulang dan dikonsolidasikan. Perubahan signifikan pada SEC 1, 4, 5, 6, 7, 8, dan 9.	27 Juni 2024
Panduan praktik terbaik yang sudah diperbarui	Praktik terbaik diperbarui dengan panduan baru di area-area berikut: Mengoperasikan beban kerja Anda dengan aman dan Melindungi data bergerak.	6 Desember 2023
Panduan praktik terbaik yang sudah diperbarui	Pembaruan besar pada panduan dan praktik terbaik di Respons insiden. Beberapa praktik terbaik diperbarui di Persiapan. Dua	3 Oktober 2023

	area baru ditambahkan ke Respons insiden: Operasi dan Aktivitas pasca-insiden. Praktik terbaik baru SEC10-BP08 Menetapkan kerangka kerja untuk belajar dari insiden sudah ditambahkan.	
Panduan praktik terbaik yang sudah diperbarui	Praktik terbaik diperbarui dengan panduan baru di area-area berikut: Persiapan dan Melakukan Simulasi.	13 Juli 2023
Pembaruan untuk Kerangka Kerja baru.	Praktik terbaik diperbarui dengan panduan preskript if dan praktik terbaik baru ditambahkan. Area praktik terbaik baru Keamanan Aplikasi (AppSec) ditambahk an.	10 April 2023
Laporan resmi diperbarui	Praktik terbaik sudah diperbarui dengan panduan implementasi yang baru.	15 Desember 2022
Laporan resmi diperbarui	Praktik terbaik diperluas dan rencana pengembangan sudah ditambahkan.	20 Oktober 2022
Pembaruan kecil	Informasi IAM diperbarui untuk menggambarkan praktik terbaik saat ini.	28 Juni 2022
Pembaruan kecil	Informasi AWS PrivateLi nk tambahan ditambahkan dan tautan yang bermasalah dikoreksi.	19 Mei 2022

Pembaruan kecil	Ditambahkan AWS PrivateLink.	6 Mei 2022
Pembaruan kecil	Bahasa noninklusif dihilangkan.	22 April 2022
Pembaruan kecil	Informasi tentang Penganalisis Akses Jaringan VPC ditambahkan.	2 Februari 2022
Pembaruan kecil	Tautan yang bermasalah diperbaiki.	27 Mei 2021
Pembaruan kecil	Perubahan editorial di seluruh dokumen.	17 Mei 2021
Pembaruan besar	Bagian tentang tata kelola ditambahkan, detail untuk berbagai bagian ditambahkan, fitur dan layanan baru ditambahkan di seluruh dokumen.	7 Mei 2021
Pembaruan kecil	Tautan diperbarui.	10 Maret 2021
Pembaruan kecil	Tautan yang bermasalah diperbaiki.	15 Juli 2020
Pembaruan untuk Kerangka Kerja baru	Pembaruan panduan untuk akun, identitas, dan manajemen izin.	8 Juli 2020
Pembaruan untuk Kerangka Kerja baru	Pembaruan untuk perluasan perangkat di setiap area, praktik terbaik baru, layanan dan fitur.	30 April 2020

Laporan resmi diperbarui	Pembaruan untuk menggambarkan fitur dan layanan AWS baru, dan pembaruan referensi.	1 Juli 2018
Laporan resmi diperbarui	Pembaruan bagian Pemeliharaan dan Konfigurasi Keamanan Sistem untuk menggambarkan fitur dan layanan AWS baru.	1 Mei 2017
Publikasi awal	Pilar Keamanan - Kerangka Kerja AWS Well-Architected diterbitkan.	1 November 2016

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik AWS produk saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. AWS produk atau layanan disediakan “sebagaimana adanya” tanpa jaminan, representasi, atau kondisi apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh AWS perjanjian, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS