Panduan Pengguna

AWS Akses Terverifikasi

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Akses Terverifikasi: Panduan Pengguna

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

Table of Contents

Apa itu Akses AWS Terverifikasi?

Dengan AWS Verifikasi, Anda dapat memberikan akses aman ke aplikasi Anda tanpa memerlukan penggunaan jaringan pribadi virtual (VPN). Akses Terverifikasi mengevaluasi setiap permintaan aplikasi dan membantu memastikan bahwa pengguna dapat mengakses setiap aplikasi hanya jika memenuhi persyaratan keamanan yang ditentukan.

Manfaat Akses Terverifikasi

- Postur keamanan yang ditingkatkan - Model keamanan tradisional mengevaluasi akses sekali dan memberi pengguna akses ke semua aplikasi. Verified Access mengevaluasi setiap permintaan akses aplikasi secara real time. Hal ini menyulitkan aktor jahat untuk berpindah dari satu aplikasi ke aplikasi lainnya.
- Integrasi dengan layanan keamanan - Akses Terverifikasi terintegrasi dengan layanan manajemen identitas dan perangkat, termasuk layanan pihak ketiga AWS dan layanan pihak ketiga. Menggunakan data dari layanan ini, Verified Access memverifikasi kepercayaan pengguna dan perangkat terhadap serangkaian persyaratan keamanan dan menentukan apakah pengguna harus memiliki akses ke aplikasi.
- Pengalaman pengguna yang ditingkatkan - Akses Terverifikasi menghilangkan kebutuhan pengguna untuk menggunakan VPN untuk mengakses aplikasi Anda. Ini membantu mengurangi jumlah kasus dukungan yang timbul dari masalah terkait VPN.
- Pemecahan masalah dan audit yang disederhanakan — Akses Terverifikasi mencatat semua upaya akses, memberikan visibilitas terpusat ke akses aplikasi, untuk membantu Anda merespons insiden keamanan dan permintaan audit dengan cepat.

Mengakses Akses AWS Terverifikasi

Anda dapat menggunakan salah satu antarmuka berikut untuk bekerja dengan Verifikasi Akses:

- AWS Management Console— Menyediakan antarmuka web untuk membuat dan mengelola sumber Verifikasi. Masuk ke AWS Management Console dan buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
- AWS Command Line Interface(AWS CLI) - Menyediakan perintah untuk serangkaian luasLayanan AWS, termasuk Akses AWS Terverifikasi. AWS CLI didukung di Windows, macOS, dan Linux. Untuk mendapatkanAWS CLI, lihat AWS Command Line Interface.

- AWSSDK — Menyediakan API khusus bahasa. AWSSDK menangani banyak detail koneksi, seperti menghitung tanda tangan, dan menangani percobaan ulang dan kesalahan. Untuk informasi selengkapnya, lihat SDK AWS.
- API tingkat rendah yang Anda hubungi menggunakan permintaan HTTPS. Menggunakan API Kueri merupakan cara paling langsung untuk mengakses Verifikasi Akses. Namun, mengharuskan aplikasi Anda menangani detail tingkat rendah seperti membuat hash untuk menandatangani permintaan dan menangani kesalahan. Untuk informasi selengkapnya, lihat Tindakan Akses Terverifikasi di Referensi API Amazon EC2.

Panduan ini menjelaskan cara menggunakan AWS Management Console untuk membuat, mengakses, dan mengelola sumber daya Akses Terverifikasi.

Harga

Anda dikenakan biaya per jam untuk setiap aplikasi di Akses Terverifikasi, dan Anda dikenai biaya untuk jumlah data yang diproses oleh Akses Terverifikasi. Untuk informasi lebih lanjut, lihat Harga AWS Verifikasi Akses.

Cara kerja Verified Access

AWSVerified Access mengevaluasi setiap permintaan aplikasi dari pengguna Anda dan memungkinkan akses berdasarkan:

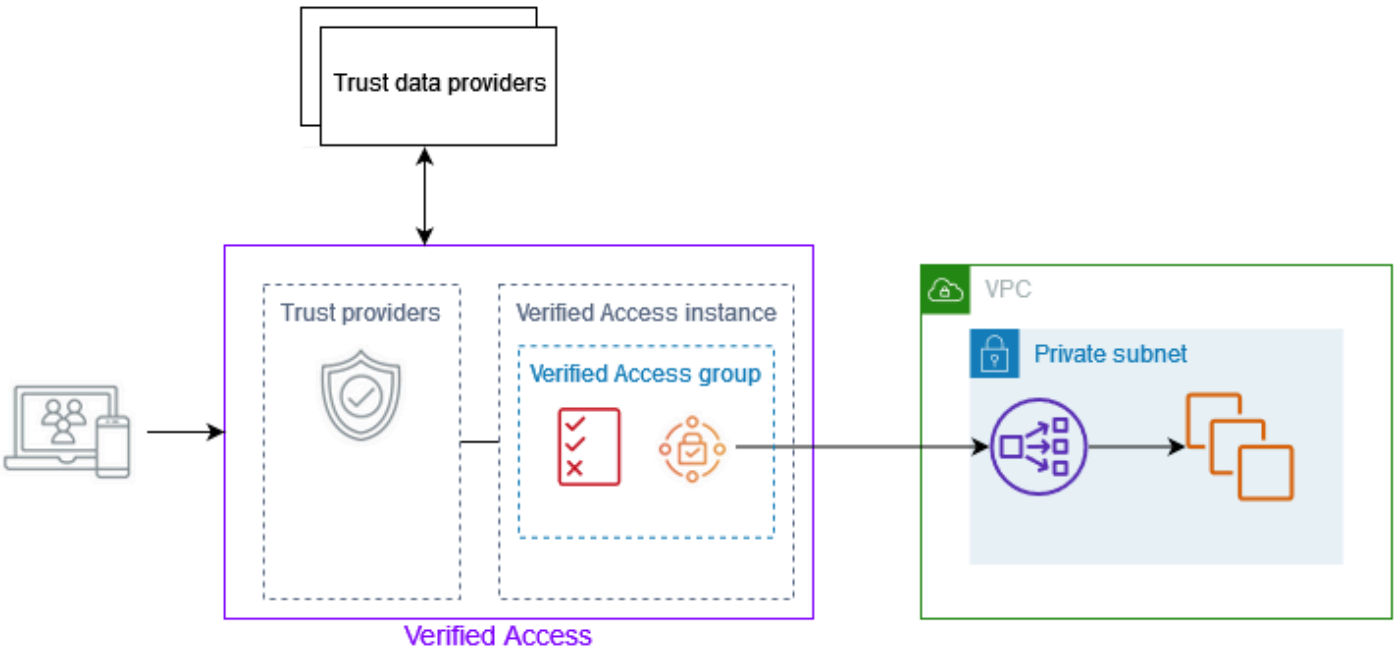
- Data kepercayaan yang dikirim oleh penyedia kepercayaan pilihan Anda (dari AWS atau pihak ketiga).
- Akses kebijakan yang Anda buat di Akses Terverifikasi.

Saat pengguna mencoba mengakses aplikasi, Akses Terverifikasi mendapatkan datanya dari penyedia kepercayaan dan mengevaluasinya terhadap kebijakan yang Anda tetapkan untuk aplikasi. Akses Terverifikasi memberikan akses ke aplikasi yang diminta hanya jika pengguna memenuhi persyaratan keamanan yang Anda tentukan. Semua permintaan aplikasi ditolak secara default, sampai kebijakan didefinisikan.

Selain itu, Akses Terverifikasi mencatat setiap upaya akses, untuk membantu Anda merespons insiden keamanan dan permintaan audit dengan cepat.

Komponen utama Akses Terverifikasi

Diagram berikut memberikan ikhtisar tingkat tinggi Akses Terverifikasi. Pengguna mengirim permintaan untuk mengakses aplikasi. Akses Terverifikasi mengevaluasi permintaan terhadap kebijakan akses untuk grup dan kebijakan titik akhir khusus aplikasi apa pun. Jika akses diizinkan, permintaan dikirim ke aplikasi melalui titik akhir.



- Instans Akses Terverifikasi — Instance mengevaluasi permintaan aplikasi dan memberikan akses hanya jika persyaratan keamanan Anda terpenuhi.
- Endpoint Akses Terverifikasi - Setiap titik akhir mewakili aplikasi. Anda dapat membuat endpoint load balancer atau endpoint antarmuka jaringan.
- Grup Akses Terverifikasi - Kumpulan titik akhir Akses Terverifikasi. Kami menyarankan Anda mengelompokkan endpoint untuk aplikasi dengan persyaratan keamanan serupa untuk menyederhanakan administrasi kebijakan. Misalnya, Anda dapat mengelompokkan titik akhir untuk semua aplikasi penjualan Anda bersama-sama.
- Kebijakan akses — Satu set aturan yang ditetapkan pengguna yang menentukan apakah akan mengizinkan atau menolak akses ke aplikasi. Anda dapat menentukan kombinasi faktor, termasuk identitas pengguna dan status keamanan perangkat. Anda membuat kebijakan akses grup untuk setiap grup Akses Terverifikasi, yang diwarisi oleh semua titik akhir dalam grup. Anda dapat membuat kebijakan khusus aplikasi secara opsional dan melampirkannya ke titik akhir tertentu.
- Penyedia kepercayaan — Layanan yang mengelola identitas pengguna atau status keamanan perangkat. Akses Terverifikasi bekerja dengan penyedia kepercayaan pihak ketiga AWS dan pihak ketiga. Anda harus melampirkan setidaknya satu penyedia kepercayaan ke setiap instans Akses Terverifikasi. Anda dapat melampirkan penyedia kepercayaan identitas tunggal dan beberapa penyedia kepercayaan perangkat ke setiap instans Akses Terverifikasi.
- Data kepercayaan — Data terkait keamanan untuk pengguna atau perangkat yang dikirim penyedia kepercayaan Anda ke Akses Terverifikasi. Juga disebut sebagai klaim pengguna atau

konteks kepercayaan. Misalnya, alamat email pengguna atau versi sistem operasi perangkat. Verified Access mengevaluasi data ini terhadap kebijakan akses Anda saat menerima setiap permintaan untuk mengakses aplikasi.

Tutorial: Memulai dengan Akses Terverifikasi

Gunakan tutorial ini untuk memulai dengan Akses AWS Terverifikasi. Anda akan mempelajari cara membuat dan mengonfigurasi sumber daya Akses Terverifikasi.

Sebelum menambahkan aplikasi ini ke Akses Terverifikasi, aplikasi hanya dapat diakses melalui jaringan pribadi Anda. Di akhir tutorial ini, pengguna tertentu dapat mengakses aplikasi yang sama melalui internet, tanpa menggunakan VPN.

> **Note**
>
> Contoh ini tidak menunjukkan integrasi dengan penyedia kepercayaan berbasis perangkat Anda. Untuk contoh ini, kami hanya bekerja dengan penyedia kepercayaan berbasis identitas.

Tugas

Prasyarat

Berikut ini adalah prasyarat untuk tutorial ini:

- Untuk menunjukkan contoh ini untuk menggunakan Akses Terverifikasi, kami akan menggunakan duaAkun AWS. Satu akun akan meng-host aplikasi target Anda, dan sumber daya Akses Terverifikasi akan dibuat di akun lain.
- Wilayah AWSAktifkan AWS IAM Identity Center di tempat Anda bekerja. Anda kemudian dapat menggunakan IAM Identity Center sebagai penyedia kepercayaan dengan Akses Terverifikasi. Untuk informasi selengkapnya, lihat Mengaktifkan Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.
- Domain yang dihosting publik dan izin yang diperlukan untuk memperbarui catatan DNS untuk domain tersebut.
- Aplikasi yang berjalan di belakang penyeimbang beban internal di fileAkun AWS. Contoh nama domain aplikasi yang akan kita gunakan adalah`www.myapp.example.com`.
- Pastikan kebijakan IAM Anda memiliki semua izin yang diperlukan untuk membuat instance Akses AWS Terverifikasi yang dicatat di sini. Kebijakan untuk membuat instance Akses Terverifikasi

Langkah 1: Buat instance Akses Terverifikasi

Gunakan prosedur berikut untuk membuat instance Akses Terverifikasi.

Untuk membuat instance Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi Amazon VPC, pilih instans Akses Terverifikasi, lalu Buat instance Akses Terverifikasi.
3. (Opsional) Untuk Nama dan Deskripsi, masukkan nama dan deskripsi untuk instance Akses Terverifikasi.
4. Untuk penyedia Trust, pertahankan opsi default.
5. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
6. Pilih Buat instance Akses Terverifikasi.

Langkah 2: Konfigurasikan penyedia kepercayaan

Anda dapat mengatur AWS IAM Identity Center sebagai penyedia kepercayaan Anda.

Untuk membuat penyedia kepercayaan Pusat Identitas IAM

1. Di panel navigasi Amazon VPC, pilih penyedia kepercayaan Akses Terverifikasi, lalu Buat penyedia kepercayaan Akses Terverifikasi.
2. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk penyedia kepercayaan Akses Terverifikasi.
3. Masukkan pengenal kustom untuk digunakan nanti saat bekerja dengan aturan kebijakan untuk nama referensi Kebijakan. Misalnya, Anda bisa masuk**`idc`**.
4. Di bawah Jenis penyedia Trust, pilih Penyedia kepercayaan pengguna.
5. Di bawah Jenis penyedia kepercayaan pengguna, pilih Pusat Identitas IAM.
6. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
7. Pilih Buat penyedia kepercayaan Akses Terverifikasi.

Langkah 3: Lampirkan penyedia kepercayaan Anda ke instans

Gunakan prosedur berikut untuk melampirkan penyedia kepercayaan ke instans Akses Terverifikasi Anda.

Untuk melampirkan penyedia kepercayaan ke instans Anda

1. Di panel navigasi Amazon VPC, pilih instans Akses Terverifikasi.
2. Pilih instans Anda.
3. Pilih Tindakan, Lampirkan penyedia kepercayaan Akses Terverifikasi.
4. Untuk penyedia kepercayaan Akses Terverifikasi, pilih penyedia kepercayaan Anda.
5. Pilih Lampirkan penyedia kepercayaan Akses Terverifikasi.

Langkah 4: Buat grup Akses Terverifikasi

Mari buat grup yang dapat Anda gunakan untuk titik akhir yang akan Anda buat di langkah berikutnya.

Untuk membuat grup Akses Terverifikasi

1. Di panel navigasi Amazon VPC, pilih grup Akses Terverifikasi, lalu Buat grup Akses Terverifikasi.

2. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk grup.
3. Untuk instance Akses Terverifikasi, pilih instans Akses Terverifikasi Anda.
4. Untuk definisi Kebijakan, kosongkan ini. Anda akan membuat kebijakan nanti dalam tutorial ini.
5. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
6. Pilih Buat grup Akses Terverifikasi.

Langkah 5: Bagikan grup Akses Terverifikasi Anda melalui AWS Resource Access Manager

Pada langkah ini, Anda akan membagikan grup yang baru saja Anda buat dengan tempat aplikasi target Anda berjalan. Akun AWS Untuk membagikan grup Akses Terverifikasi, Anda harus menambahkannya ke pembagian sumber daya. Jika Anda tidak memiliki pembagian sumber daya, Anda harus membuatnya terlebih dahulu.

Jika Anda adalah bagian dari organisasi diAWS Organizations, dan berbagi dalam organisasi Anda diaktifkan, konsumen di organisasi Anda secara otomatis diberikan akses ke grup Akses Terverifikasi bersama. Jika tidak, konsumen akan menerima undangan untuk bergabung dengan pembagian sumber daya dan diberikan akses ke grup Akses Terverifikasi bersama setelah menerima undangan.

Ikuti langkah-langkah di Buat pembagian sumber daya di Panduan Pengguna AWS RAM. Untuk Pilih jenis sumber daya, pilih grup Akses Terverifikasi, lalu pilih kotak centang untuk grup Akses Terverifikasi Anda.

Untuk informasi selengkapnya, lihat Memulai di Panduan AWS RAM Pengguna.

Langkah 6: Tambahkan aplikasi Anda dengan membuat titik akhir

Gunakan prosedur berikut untuk membuat titik akhir. Langkah ini mengasumsikan bahwa Anda memiliki aplikasi yang berjalan di belakang penyeimbang beban internal dari Elastic Load Balancing.

Untuk membuat titik akhir Akses Terverifikasi

1. Di panel navigasi Amazon VPC, pilih titik akhir Akses Terverifikasi, lalu Buat titik akhir Akses Terverifikasi.
2. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk titik akhir.

3. Untuk grup Akses Terverifikasi, pilih grup Akses Terverifikasi Anda.
4. Untuk detail Aplikasi, lakukan hal berikut:
   a. Untuk domain Aplikasi, masukkan nama DNS untuk aplikasi Anda.
   b. Di bawah Sertifikat domain ARN, pilih Nama Sumber Daya Amazon (ARN) sertifikat TLS publik Anda.
5. Untuk detail Endpoint, lakukan hal berikut:
   a. Untuk jenis Lampiran, pilih VPC.
   b. Untuk grup Keamanan, pilih grup keamanan untuk dikaitkan dengan titik akhir.
   c. Untuk awalan domain Endpoint, masukkan pengenal kustom. Ini akan ditambahkan ke nama DNS yang dihasilkan Akses Terverifikasi. Untuk contoh ini, kita bisa menggunakan**`my-ava-app`**.
   d. Untuk tipe Endpoint, pilih Load balancer.
   e. Untuk Protokol, pilih HTTPS atau HTTP. Ini tergantung pada konfigurasi penyeimbang beban Anda.
   f. Untuk Port, masukkan nomor port. Ini tergantung pada konfigurasi penyeimbang beban Anda.
   g. Untuk Load balancer ARN, pilih load balancer Anda.
   h. Untuk Subnet, pilih subnet yang terkait dengan penyeimbang beban Anda.
6. Untuk definisi Kebijakan, jangan masukkan kebijakan saat ini. Kami akan membahas ini nanti di tutorial.
7. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
8. Pilih Buat titik akhir Akses Terverifikasi.

Langkah 7: Konfigurasikan pengaturan DNS

Untuk langkah ini, Anda memetakan nama domain aplikasi Anda (misalnya, www.myapp.example.com) ke nama domain titik akhir Akses Terverifikasi Anda. Untuk menyelesaikan pemetaan DNS, buat Canonical Name Record (CNAME) dengan penyedia DNS Anda. Setelah Anda membuat catatan CNAME, semua permintaan dari pengguna ke aplikasi Anda akan dikirim ke Akses Terverifikasi.

Untuk mendapatkan nama domain dari endpoint Anda

1. Di panel navigasi Amazon VPC, pilih titik akhir Akses Terverifikasi.
2. Pilih titik akhir yang Anda buat sebelumnya.
3. Pilih tab Detail untuk titik akhir.
4. Salin domain endpoint dari bawah domain Endpoint.

Untuk tutorial ini, nama domain endpoint akan menjadi`my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`.

Buat catatan CNAME dengan penyedia DNS Anda:

Nama catatan	Tipe	Nilai
www.myapp.example.com	CNAME	my-ava-app.edge-1a 2b3c4d5e6f7g.vai-1 a2b3c4d5e6f7g.prod. verified-access.us-west-2.amazonaws. com

Langkah 8: Uji konektivitas ke aplikasi Anda

Anda sekarang dapat menguji konektivitas ke aplikasi Anda. Masukkan nama domain aplikasi Anda ke browser web Anda. Perilaku default kebijakan Akses Terverifikasi adalah menolak semua permintaan. Karena kami belum menerapkan kebijakan yang memungkinkan siapa pun mengakses, semua permintaan harus ditolak.

Langkah 9: Konfigurasikan kebijakan akses tingkat grup

Gunakan prosedur berikut untuk mengubah grup Akses Terverifikasi dan mengonfigurasi kebijakan akses yang memungkinkan konektivitas ke aplikasi Anda. Rincian kebijakan akan tergantung pada pengguna dan grup yang dikonfigurasi di Pusat Identitas IAM. Untuk informasi tentang membuat kebijakan, lihatKebijakan Akses Terverifikasi.

Untuk mengubah grup Akses Terverifikasi

1. Di panel navigasi Amazon VPC, pilih grup Akses Terverifikasi.
2. Pilih grup Anda.
3. Pilih Tindakan, Ubah kebijakan grup Akses Terverifikasi.
4. Masukkan kebijakan.
5. Pilih Ubah kebijakan grup Akses Terverifikasi.

Langkah 10: Uji ulang konektivitas

Sekarang setelah kebijakan grup Anda diberlakukan, Anda dapat mengakses aplikasi Anda. Masukkan nama domain aplikasi Anda ke browser web Anda. Permintaan harus diizinkan dan Anda harus diarahkan ke aplikasi.

Hapus

Setelah Anda selesai menguji, ikuti langkah di bawah ini untuk menghapus sumber daya yang dibuat.

Untuk menghapus sumber daya Akses Terverifikasi yang dibuat dengan tutorial ini

1. Di panel navigasi Amazon VPC, pilih titik akhir Akses Terverifikasi. Pilih titik akhir yang ingin Anda hapus. Pilih Tindakan, Hapus titik akhir Akses Terverifikasi.
2. Di panel navigasi, pilih grup Akses Terverifikasi. Pilih grup yang ingin Anda hapus. Pilih Tindakan, Hapus grup Akses Terverifikasi. Catatan - Anda mungkin perlu menunggu beberapa menit hingga proses penghapusan titik akhir selesai.
3. Di panel navigasi Amazon VPC, pilih instans Akses Terverifikasi. Pilih contoh yang Anda buat untuk tutorial ini. Pilih Tindakan, Lepaskan penyedia kepercayaan Akses Terverifikasi. Pilih penyedia kepercayaan dari daftar drop-down, pilih Lepaskan penyedia kepercayaan Akses Terverifikasi.
4. Di panel navigasi Amazon VPC, pilih penyedia kepercayaan Akses Terverifikasi. Pilih penyedia kepercayaan yang Anda buat untuk tutorial ini. Pilih Tindakan, Hapus penyedia kepercayaan Akses Terverifikasi.
5. Di panel navigasi Amazon VPC, pilih instans Akses Terverifikasi. Pilih contoh yang Anda buat untuk tutorial ini. Pilih Tindakan, Hapus instans Akses Terverifikasi.

Instans Akses Terverifikasi

Instans Akses AWS Terverifikasi adalah AWS sumber daya yang membantu Anda mengatur penyedia kepercayaan dan grup Akses Terverifikasi.

Topik

Buat instance Akses Terverifikasi

Gunakan prosedur berikut untuk membuat instance Akses Terverifikasi.

Untuk membuat instance Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instance Akses Terverifikasi, lalu Buat instance Akses Terverifikasi.
3. (Opsional) Untuk Nama dan Deskripsi, masukkan nama dan deskripsi untuk instance Akses Terverifikasi.
4. (Opsional) Pilih aktifkan untuk Standar Proses Informasi Federal (FIPS) jika Anda memerlukan Akses Terverifikasi agar sesuai dengan FIPS.
5. (Opsional) Untuk penyedia Trust, pilih penyedia kepercayaan untuk dilampirkan ke instance Akses Terverifikasi.
6. (Opsional) Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag.
7. Pilih Buat instance Akses Terverifikasi.

Lampirkan penyedia kepercayaan ke sebuah instans

Gunakan prosedur berikut untuk melampirkan penyedia kepercayaan ke instance Akses Terverifikasi.

Untuk melampirkan penyedia kepercayaan ke instans Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instans.
4. Pilih Tindakan, Lampirkan penyedia kepercayaan Akses Terverifikasi.
5. Untuk penyedia kepercayaan Akses Terverifikasi, pilih penyedia kepercayaan.
6. Pilih Lampirkan penyedia kepercayaan Akses Terverifikasi.

Lepaskan penyedia kepercayaan dari sebuah instans

Gunakan prosedur berikut untuk melepaskan penyedia kepercayaan dari instance Akses Terverifikasi.

Untuk melepaskan penyedia kepercayaan dari instans Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instans.
4. Pilih Tindakan, Lepaskan penyedia kepercayaan Akses Terverifikasi.
5. Untuk penyedia kepercayaan Akses Terverifikasi, pilih penyedia kepercayaan.
6. Pilih Lepaskan penyedia kepercayaan Akses Terverifikasi.

Menghapus instans Akses Terverifikasi

Setelah selesai dengan instance Akses Terverifikasi, Anda dapat menghapusnya. Sebelum menghapus instans, Anda harus menghapus penyedia kepercayaan terkait atau grup Akses Terverifikasi.

Untuk menghapus instans Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi.

4. Pilih Tindakan, Hapus instans Akses Terverifikasi.
5. Saat diminta konfirmasi, masukkan**`delete`**, lalu pilih Hapus.

Mengintegrasikan dengan AWS WAF

Selain aturan otentikasi dan otorisasi yang diberlakukan oleh Akses Terverifikasi, Anda mungkin juga ingin menerapkan perlindungan perimeter. Ini dapat membantu Anda melindungi aplikasi Anda dari ancaman tambahan. Anda dapat melakukannya dengan mengintegrasikan AWS WAF ke dalam penerapan Akses Terverifikasi Anda. AWS WAFadalah firewall aplikasi web yang memungkinkan Anda memantau permintaan HTTP (S) yang diteruskan ke sumber daya aplikasi web Anda yang dilindungi. Untuk informasi selengkapnyaAWS WAF, lihat AWS WAFdi Panduan AWS WAF Pengembang.

Anda dapat mengintegrasikan AWS WAF dengan Akses Terverifikasi dengan mengaitkan daftar kontrol akses AWS WAF web (ACL) dengan instans Akses Terverifikasi. ACL web adalah AWS WAF sumber daya yang memberi Anda kontrol halus atas semua permintaan web HTTP (S) yang ditanggapi oleh sumber daya terlindungi Anda. Saat permintaan AWS WAF asosiasi atau disasosiasi sedang diproses, status titik akhir Akses Terverifikasi yang dilampirkan ke instance ditampilkan sebagai. `updating` Setelah permintaan selesai, status kembali `keactive`. Anda dapat melihat status di AWS Management Console atau dengan menjelaskan titik akhir dengan. AWS CLI

> **Note**
>
> Anda juga dapat menggunakan AWS WAF konsol atau API untuk menyelesaikan integrasi ini. Anda akan memerlukan Nama Sumber Daya Amazon (ARN) dari instans Akses Terverifikasi Anda. Anda dapat membangun ARN ini menggunakan format berikut:. `arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}`

Topik

- Izin IAM diperlukan untuk mengintegrasikan AWS WAF
- Kaitkan ACL AWS WAF web
- Periksa status AWS WAF integrasi
- Putuskan hubungan ACL AWS WAF web

Izin IAM diperlukan untuk mengintegrasikan AWS WAF

Mengintegrasikan AWS WAF dengan Akses Terverifikasi mencakup tindakan khusus izin yang tidak berhubungan langsung dengan operasi API. Tindakan ini ditunjukkan dalam Referensi Otorisasi AWS Identity and Access Management Layanan dengan`[permission only]`. Lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon EC2 di Referensi Otorisasi Layanan.

Untuk bekerja dengan ACL web, AWS Identity and Access Management kepala sekolah Anda harus memiliki izin berikut.

- `ec2:AssociateVerifiedAccessInstanceWebAcl`
- `ec2:DisassociateVerifiedAccessInstanceWebAcl`
- `ec2:DescribeVerifiedAccessInstanceWebAclAssociations`
- `ec2:GetVerifiedAccessInstanceWebAcl`

Kaitkan ACL AWS WAF web

Langkah-langkah berikut menunjukkan cara mengaitkan daftar kontrol akses AWS WAF web (ACL) dengan instance Akses Terverifikasi menggunakan. AWS Management Console

> **Tip**
>
> Anda harus memiliki ACL AWS WAF web yang ada untuk menyelesaikan prosedur di bawah ini. Untuk informasi selengkapnya tentang ACL web, lihat daftar kontrol akses Web di Panduan AWS WAF Pengembang.

Untuk mengaitkan ACL AWS WAF web ke instans Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi.
4. Pilih tab Integrasi.
5. Pilih Actions, lalu Associate Web ACL.
6. Untuk Web ACL, pilih ACL web yang ada, lalu pilih Associate Web ACL.

Anda juga dapat menggunakan AWS Management Console for AWS WAF untuk menyelesaikan tugas ini. Untuk informasi selengkapnya, lihat Mengaitkan atau memisahkan ACL web dengan sumber daya AWS di Panduan Pengembang. AWS WAF

Periksa status AWS WAF integrasi

Anda dapat memverifikasi apakah daftar kontrol akses AWS WAF web (ACL) dikaitkan dengan instance Akses Terverifikasi atau tidak dengan menggunakan. AWS Management Console

Untuk melihat status AWS WAF integrasi dengan instans Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi.
4. Pilih tab Integrasi.
5. Periksa detail yang tercantum di bawah status integrasi WAF. Status akan ditampilkan sebagai Terkait atau Tidak terkait, bersama dengan pengenal ACL web, jika dalam status Terkait.

Putuskan hubungan ACL AWS WAF web

Langkah-langkah berikut menunjukkan cara memisahkan daftar kontrol akses AWS WAF web (ACL) dengan instance Akses Terverifikasi menggunakan. AWS Management Console

Untuk memisahkan ACL AWS WAF web dari instans Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi.
4. Pilih tab Integrasi.
5. Pilih Tindakan, lalu Putuskan Web ACL.
6. Konfirmasikan dengan memilih Disassociate Web ACL.

Anda juga dapat menggunakan AWS Management Console for AWS WAF untuk menyelesaikan tugas ini. Untuk informasi selengkapnya, lihat Mengaitkan atau memisahkan ACL web dengan sumber daya AWS di Panduan Pengembang. AWS WAF

Kepatuhan FIPS untuk Akses Terverifikasi

Federal Information Processing Standard (FIPS) adalah standar pemerintah AS dan Kanada yang menetapkan persyaratan keamanan untuk modul kriptografi yang melindungi informasi sensitif. Akses Terverifikasi AWSmenyediakan opsi untuk mengonfigurasi lingkungan Anda agar mematuhi Publikasi FIPS 140-2. Kepatuhan FIPS untuk Akses Terverifikasi tersedia di AWS Wilayah berikut:

- AS Timur (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Canada (Central)

Halaman ini menunjukkan cara mengonfigurasi lingkungan Akses Terverifikasi baru, atau yang sudah ada, agar sesuai dengan FIPS.

Topik

Konfigurasikan lingkungan Akses Terverifikasi yang ada untuk kepatuhan FIPS

Jika Anda memiliki lingkungan Akses Terverifikasi yang ada dan Anda ingin mengonfigurasinya agar sesuai dengan FIPS, beberapa sumber daya perlu dihapus dan dibuat ulang untuk mengaktifkan kepatuhan FIPS.

Untuk mengonfigurasi ulang Akses Terverifikasi AWS lingkungan yang ada agar sesuai dengan FIPS, ikuti langkah-langkah di bawah ini.

1. Hapus titik akhir, grup, dan instans Akses Terverifikasi asli Anda. Penyedia kepercayaan Anda yang dikonfigurasi dapat digunakan kembali.
2. Buat instance Akses Terverifikasi, pastikan untuk mengaktifkan Standar Proses Informasi Federal (FIPS) selama pembuatan. Juga selama pembuatan, lampirkan penyedia kepercayaan Akses Terverifikasi yang ingin Anda gunakan, dengan memilihnya dari daftar drop-down.

3. Buat grup Akses Terverifikasi. Selama pembuatan grup, Anda mengaitkannya dengan instance Akses Terverifikasi yang baru saja dibuat.
4. Buat satu atau lebihTitik akhir Akses Terverifikasi. Selama pembuatan titik akhir Anda, Anda mengaitkannya dengan grup yang dibuat pada langkah sebelumnya.

Konfigurasikan lingkungan Akses Terverifikasi baru untuk kepatuhan FIPS

Untuk mengonfigurasi Akses Terverifikasi AWS lingkungan baru yang sesuai dengan FIPS, ikuti langkah-langkah di bawah ini.

1. Konfigurasikan penyedia kepercayaan. Anda perlu membuat penyedia kepercayaan identitas pengguna dan (opsional) penyedia kepercayaan berbasis perangkat, tergantung pada kebutuhan Anda.
2. Buat instance Akses Terverifikasi, pastikan untuk mengaktifkan Standar Proses Informasi Federal (FIPS) selama proses berlangsung. Juga selama pembuatan, lampirkan penyedia kepercayaan Akses Terverifikasi yang Anda buat di langkah sebelumnya, dengan memilihnya dari daftar drop-down.
3. Buat grup Akses Terverifikasi. Selama pembuatan grup, Anda mengaitkannya dengan instance Akses Terverifikasi yang baru saja dibuat.
4. Buat satu atau lebihTitik akhir Akses Terverifikasi. Selama pembuatan titik akhir Anda, Anda mengaitkannya dengan grup yang dibuat pada langkah sebelumnya.

Penyedia kepercayaan untuk Akses Terverifikasi

Penyedia kepercayaan adalah layanan yang mengirimkan informasi tentang pengguna dan perangkat ke Akses AWS Terverifikasi. Informasi ini disebut konteks kepercayaan. Ini dapat mencakup atribut berdasarkan identitas pengguna, seperti alamat email atau keanggotaan dalam organisasi “penjualan”, atau informasi perangkat seperti patch keamanan yang diinstal atau versi perangkat lunak anti-virus.

Akses Terverifikasi mendukung kategori penyedia kepercayaan berikut:

- Identitas pengguna — Layanan penyedia identitas (iDP) yang menyimpan dan mengelola identitas digital untuk pengguna.
- Manajemen perangkat — Sistem manajemen perangkat untuk perangkat seperti laptop, tablet, dan smartphone.

Daftar Isi

Penyedia kepercayaan identitas pengguna

Anda dapat memilih untuk menggunakan salah satu AWS IAM Identity Center atau penyedia kepercayaan identitas pengguna yang kompatibel dengan OpenID Connect.

Daftar Isi

Menggunakan IAM Identity Center sebagai penyedia kepercayaan

Anda dapat menggunakan AWS IAM Identity Center sebagai penyedia kepercayaan identitas pengguna dengan Akses AWS Terverifikasi.

Prasyarat dan pertimbangan

- Instance IAM Identity Center Anda harus berupa sebuah AWS Organizations instance. Instans Pusat Identitas IAM AWS akun mandiri tidak akan berfungsi.
- Instance Pusat Identitas IAM Anda harus diaktifkan di AWS Wilayah yang sama tempat Anda ingin membuat penyedia kepercayaan Akses Terverifikasi.

Lihat Mengelola instans organisasi dan akun Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna untuk detail tentang berbagai jenis instans.

Buat penyedia kepercayaan Pusat Identitas IAM

Setelah Pusat Identitas IAM diaktifkan di AWS akun Anda, Anda dapat menggunakan prosedur berikut untuk menyiapkan Pusat Identitas IAM sebagai penyedia kepercayaan Anda untuk Akses Terverifikasi.

Untuk membuat penyedia kepercayaan Pusat Identitas IAM (AWSkonsol)

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi, lalu Buat penyedia kepercayaan Akses Terverifikasi.
3. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk penyedia kepercayaan.
4. Untuk nama referensi Kebijakan, masukkan pengenal yang akan digunakan nanti saat bekerja dengan aturan kebijakan.
5. Di bawah Jenis penyedia Trust, pilih Penyedia kepercayaan pengguna.
6. Di bawah Jenis penyedia kepercayaan pengguna, pilih Pusat Identitas IAM.
7. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
8. Pilih Buat penyedia kepercayaan Akses Terverifikasi.

Untuk membuat penyedia kepercayaan Pusat Identitas IAM (AWSCLI)

- create-verified-access-trust-penyedia () AWS CLI

Hapus penyedia kepercayaan Pusat Identitas IAM

Sebelum Anda dapat menghapus penyedia kepercayaan, Anda harus menghapus semua konfigurasi titik akhir dan grup dari instance yang dilampirkan penyedia kepercayaan.

Untuk menghapus penyedia kepercayaan Pusat Identitas IAM (AWSkonsol)

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi, lalu pilih penyedia kepercayaan yang ingin Anda hapus di bawah Penyedia kepercayaan Akses Terverifikasi.
3. Pilih Tindakan, lalu Hapus penyedia kepercayaan Akses Terverifikasi.
4. Konfirmasikan penghapusan dengan memasukkan `delete` ke dalam kotak teks.
5. Pilih Hapus.

Untuk menghapus penyedia kepercayaan Pusat Identitas IAM (AWSCLI)

- delete-verified-access-trust-penyedia () AWS CLI

Menggunakan penyedia kepercayaan OpenID Connect

AWSVerified Access mendukung penyedia identitas yang menggunakan metode OpenID Connect (OIDC) standar. Anda dapat menggunakan penyedia yang kompatibel dengan OIDC sebagai penyedia kepercayaan identitas pengguna dengan Akses Terverifikasi. Namun, karena beragam penyedia OIDC potensial, AWS tidak dapat menguji setiap integrasi OIDC dengan Akses Terverifikasi.

Akses Terverifikasi memperoleh data kepercayaan yang dievaluasi dari penyedia OIDC. `UserInfo Endpoint` `Scope`Parameter ini digunakan untuk menentukan kumpulan data kepercayaan mana yang akan diambil. Setelah data kepercayaan diterima, kebijakan Akses Terverifikasi dievaluasi terhadapnya.

> **Note**
>
> Akses Terverifikasi tidak menggunakan data kepercayaan dari yang `ID token` dikirim oleh penyedia OIDC, saat mengevaluasi kebijakan Akses Terverifikasi. Hanya data kepercayaan dari yang `UserInfo Endpoint` dievaluasi terhadap kebijakan.

Daftar Isi

Prasyarat untuk membuat penyedia kepercayaan OIDC

Anda perlu mengumpulkan informasi berikut dari layanan penyedia kepercayaan Anda secara langsung:

- Penerbit
- Titik akhir otorisasi
- Titik akhir token
- UserInfo titik akhir
- ID Klien
- Rahasia klien
- Cakupan

Buat penyedia kepercayaan OIDC

Gunakan prosedur berikut untuk membuat OIDC sebagai penyedia kepercayaan Anda.

Untuk membuat penyedia kepercayaan OIDC (konsol) AWS

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi, lalu Buat penyedia kepercayaan Akses Terverifikasi.
3. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk penyedia kepercayaan.
4. Untuk nama referensi Kebijakan, masukkan pengenal yang akan digunakan nanti saat bekerja dengan aturan kebijakan.
5. Di bawah Jenis penyedia Trust, pilih Penyedia kepercayaan pengguna.
6. Di bawah Jenis penyedia kepercayaan pengguna, pilih OIDC (OpenID Connect).

7. Untuk Penerbit, masukkan pengenal penerbit OIDC.
8. Untuk titik akhir Otorisasi, masukkan URL lengkap titik akhir otorisasi.
9. Untuk titik akhir Token, masukkan URL lengkap titik akhir token.
10. Untuk titik akhir Pengguna, masukkan URL lengkap titik akhir pengguna.
11. Masukkan pengenal klien OAuth 2.0 untuk ID Klien.
12. Masukkan rahasia klien OAuth 2.0 untuk rahasia Klien.
13. Masukkan daftar cakupan yang dibatasi spasi yang ditentukan dengan penyedia identitas Anda. Minimal, lingkup “openid” diperlukan untuk Lingkup.
14. (Opsional) Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
15. Pilih Buat penyedia kepercayaan Akses Terverifikasi.

> **Note**
>
> Anda perlu menambahkan URI pengalihan ke daftar izin penyedia OIDC Anda. Anda akan ingin menggunakan titik akhir Akses Terverifikasi untuk tujuan ini. `ApplicationDomain` Ini dapat ditemukan diAWS Management Console, di bawah tab Detail untuk titik akhir Akses Terverifikasi Anda atau dengan menggunakan AWS CLI untuk menggambarkan titik akhir. Tambahkan yang berikut ini ke daftar izin penyedia OIDC Anda: https:///oauth2/idpresponse `ApplicationDomain`

Untuk membuat penyedia kepercayaan OIDC (CLIAWS)

- create-verified-access-trust-penyedia () AWS CLI

Memodifikasi penyedia kepercayaan OIDC

Setelah Anda membuat penyedia kepercayaan, Anda dapat memperbarui konfigurasinya.

Untuk memodifikasi penyedia kepercayaan OIDC (konsol) AWS

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi, lalu pilih penyedia kepercayaan yang ingin Anda ubah di bawah Penyedia kepercayaan Akses Terverifikasi.

3. Pilih Tindakan, lalu Ubah penyedia kepercayaan Akses Terverifikasi.
4. Ubah opsi yang ingin Anda ubah.
5. Pilih Ubah penyedia kepercayaan Akses Terverifikasi.

Untuk memodifikasi penyedia kepercayaan OIDC (CLIAWS)

- modify-verified-access-trust-penyedia () AWS CLI

Hapus penyedia kepercayaan OIDC

Sebelum dapat menghapus penyedia kepercayaan pengguna, pertama-tama Anda harus menghapus semua konfigurasi titik akhir dan grup dari contoh penyedia kepercayaan yang dilampirkan.

Untuk menghapus penyedia kepercayaan OIDC (konsol) AWS

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi, lalu pilih penyedia kepercayaan yang ingin Anda hapus di bawah Penyedia kepercayaan Akses Terverifikasi.
3. Pilih Tindakan, lalu Hapus penyedia kepercayaan Akses Terverifikasi.
4. Konfirmasikan penghapusan dengan memasukkan `delete` ke dalam kotak teks.
5. Pilih Hapus.

Untuk menghapus penyedia kepercayaan OIDC (CLIAWS)

- delete-verified-access-trust-penyedia () AWS CLI

Penyedia kepercayaan berbasis perangkat

Anda dapat menggunakan penyedia kepercayaan perangkat dengan Akses AWS Terverifikasi. Anda dapat menggunakan satu atau beberapa penyedia kepercayaan perangkat dengan instans Akses Terverifikasi.

Daftar Isi

- Memodifikasi penyedia kepercayaan berbasis perangkat
- Menghapus penyedia kepercayaan berbasis perangkat

Penyedia kepercayaan perangkat yang didukung

Penyedia kepercayaan perangkat berikut dapat diintegrasikan dengan Akses Terverifikasi:

- CrowdStrike — Mengamankan aplikasi pribadi dengan CrowdStrike dan Akses Terverifikasi
- Jamf - Mengintegrasikan Akses Terverifikasi dengan Identitas Perangkat Jamf
- JumpCloud — Mengintegrasikan JumpCloud dan Akses AWS Terverifikasi

Buat penyedia kepercayaan berbasis perangkat

Ikuti langkah-langkah berikut untuk membuat dan mengonfigurasi penyedia kepercayaan perangkat untuk digunakan dengan Akses Terverifikasi.

Untuk membuat penyedia kepercayaan perangkat Akses Terverifikasi (AWSkonsol)

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi, lalu Buat penyedia kepercayaan Akses Terverifikasi.
3. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk penyedia kepercayaan.
4. Masukkan pengenal untuk digunakan nanti saat bekerja dengan aturan kebijakan untuk nama referensi Kebijakan.
5. Untuk jenis penyedia Trust, pilih Identitas perangkat.
6. Untuk jenis identitas Perangkat, pilih Jamf, CrowdStrike, atau JumpCloud.
7. Untuk ID Penyewa, masukkan pengidentifikasi aplikasi penyewa.
8. (Opsional) Untuk URL kunci penandatanganan publik, masukkan URL kunci unik yang dibagikan oleh penyedia kepercayaan perangkat Anda. (Parameter ini tidak diperlukan untuk Jamf, CrowdStrike atau Jumpcloud.)
9. Pilih Buat penyedia kepercayaan Akses Terverifikasi.

> **Note**
>
> Anda perlu menambahkan URI pengalihan ke daftar izin penyedia OIDC Anda. Anda akan ingin menggunakan titik akhir Akses Terverifikasi untuk tujuan ini. `DeviceValidationDomain` Ini dapat ditemukan diAWS Management Console, di bawah tab Detail untuk titik akhir Akses Terverifikasi Anda atau dengan menggunakan AWS CLI untuk menggambarkan titik akhir. Tambahkan yang berikut ini ke daftar izin penyedia OIDC Anda: https:///oauth2/idpresponse `DeviceValidationDomain`

Untuk membuat penyedia kepercayaan perangkat Akses Terverifikasi (AWSCLI)

- create-verified-access-trust-penyedia () AWS CLI

Memodifikasi penyedia kepercayaan berbasis perangkat

Setelah Anda membuat penyedia kepercayaan, Anda dapat memperbarui konfigurasinya.

Untuk mengubah penyedia kepercayaan perangkat Akses Terverifikasi (AWSkonsol)

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi.
3. Pilih penyedia kepercayaan.
4. Pilih Tindakan, lalu pilih Ubah penyedia kepercayaan Akses Terverifikasi.
5. Ubah deskripsi sesuai kebutuhan.
6. (Opsional) Untuk URL kunci penandatanganan publik, ubah URL kunci unik yang dibagikan oleh penyedia kepercayaan perangkat Anda. (Parameter ini tidak diperlukan jika penyedia kepercayaan perangkat Anda adalah Jamf, CrowdStrike atau Jumpcloud.)
7. Pilih Ubah penyedia kepercayaan Akses Terverifikasi.

Untuk mengubah penyedia kepercayaan perangkat Akses Terverifikasi (AWSCLI)

- modify-verified-access-trust-penyedia () AWS CLI

Menghapus penyedia kepercayaan berbasis perangkat

Setelah selesai dengan penyedia kepercayaan, Anda dapat menghapusnya.

Untuk menghapus penyedia kepercayaan perangkat Akses Terverifikasi (AWSkonsol)

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Penyedia kepercayaan Akses Terverifikasi.
3. Pilih penyedia kepercayaan yang ingin Anda hapus di bawah Penyedia kepercayaan Akses Terverifikasi.
4. Pilih Tindakan, lalu pilih Hapus penyedia kepercayaan Akses Terverifikasi.
5. Saat diminta konfirmasi, masukkan**`delete`**, lalu pilih Hapus.

Untuk menghapus penyedia kepercayaan perangkat Akses Terverifikasi (AWSCLI)

- delete-verified-access-trust-penyedia () AWS CLI

Grup Akses Verifikasi

Grup Akses AWS Terverifikasi adalah kumpulan titik akhir Akses Terverifikasi dan kebijakan Akses Terverifikasi tingkat grup. Setiap titik akhir dalam grup berbagi kebijakan Akses Terverifikasi. Anda dapat menggunakan grup untuk mengumpulkan titik akhir yang memiliki persyaratan keamanan umum. Ini dapat membantu menyederhanakan administrasi kebijakan dengan menggunakan satu kebijakan untuk kebutuhan keamanan beberapa aplikasi.

Misalnya, Anda dapat mengelompokkan semua aplikasi penjualan bersama-sama dan menetapkan kebijakan akses seluruh grup. Anda kemudian dapat menggunakan kebijakan ini untuk menentukan serangkaian persyaratan keamanan minimum yang umum untuk semua aplikasi penjualan. Pendekatan ini membantu menyederhanakan administrasi kebijakan.

Saat Anda membuat grup, Anda diminta untuk menghubungkan grup tersebut dengan instans Verified Access. Selama proses pembuatan endpoint, Anda akan mengaitkan titik akhir dengan grup.

Tugas

- Buat grup Akses Terverifikasi
- Memodifikasi kebijakan grup Akses Terverifikasi
- Menghapus grup Akses Terverifikasi

Buat grup Akses Terverifikasi

Gunakan prosedur berikut untuk membuat grup Verified Access.

Membuat grup Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Grup Akses Terverifikasi, lalu Buat Grup Akses Terverifikasi.
3. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk grup.
4. Untuk instance Akses Terverifikasi, pilih instance Akses Terverifikasi untuk dikaitkan dengan grup.
5. (Opsional) Untuk definisi Kebijakan, masukkan kebijakan Akses Terverifikasi untuk diterapkan ke grup.
6. (Opsional) Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan kunci dan nilai tag tersebut.

7. Pilih Buat grup Akses Terverifikasi.

Memodifikasi kebijakan grup Akses Terverifikasi

Gunakan prosedur berikut untuk memodifikasi kebijakan grup Verified Access.

Mengubah kebijakan grup Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih grup Akses Verifikasi, lalu pilih grup yang kebijakannya ingin Anda modifikasi.
3. Pilih Tindakan, lalu Ubah kebijakan grup Akses Terverifikasi.
4. (Opsional) Aktifkan atau nonaktifkan Aktifkan kebijakan tergantung pada tujuan Anda saat ini.
5. (Opsional) Untuk Kebijakan, masukkan kebijakan Akses Terverifikasi untuk diterapkan ke grup.
6. Pilih Ubah kebijakan grup Akses Terverifikasi.

Menghapus grup Akses Terverifikasi

Setelah Anda selesai dengan grup Verified Access, Anda dapat menghapusnya.

Untuk menghapus grup Akses Verifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih Grup Akses Verifikasi.
3. Pilih grup .
4. Pilih Tindakan, Hapus Grup Akses Terverifikasi.
5. Ketika diminta untuk mengonfirmasi, masukkan**`delete`**, lalu pilih Hapus.

Titik akhir Akses Terverifikasi

Titik akhir Akses Terverifikasi mewakili aplikasi. Setiap titik akhir dikaitkan dengan grup Akses Terverifikasi dan mewarisi kebijakan akses untuk grup. Anda dapat melampirkan kebijakan endpoint khusus aplikasi secara opsional ke setiap titik akhir.

Daftar Isi

Jenis titik akhir Akses Terverifikasi

Berikut ini adalah jenis titik akhir yang mungkin:

- Load balancer — Permintaan aplikasi dikirim ke penyeimbang beban untuk didistribusikan ke aplikasi Anda.
- Antarmuka jaringan — Permintaan aplikasi dikirim ke antarmuka jaringan menggunakan protokol dan port yang ditentukan.

VPC dan subnet bersama

Berikut ini adalah perilaku terkait subnet VPC bersama:

- Titik akhir Akses Terverifikasi didukung oleh berbagi subnet VPC. Peserta dapat membuat titik akhir Akses Terverifikasi di subnet bersama.
- Peserta yang membuat endpoint akan menjadi pemilik endpoint, dan satu-satunya pihak yang diizinkan untuk memodifikasi endpoint. Pemilik VPC tidak akan diizinkan untuk memodifikasi titik akhir.

- Titik akhir Akses Terverifikasi tidak dapat dibuat di Zona AWS Lokal dan oleh karena itu berbagi melalui Local Zones tidak dimungkinkan.

Untuk informasi selengkapnya, lihat, Bagikan VPC Anda dengan akun lain di Panduan Pengguna Amazon VPC.

Membuat titik akhir penyeimbang beban untuk Akses Terverifikasi

Gunakan prosedur berikut untuk membuat titik akhir penyeimbang beban. Untuk informasi selengkapnya tentang load balancer, lihat Panduan Pengguna Elastic Load Balancing.

Persyaratan

- Hanya lalu lintas IPv4 yang didukung.
- Hanya protokol HTTP dan HTTPS yang didukung.
- Load balancer harus berupa Application Load Balancer atau Network Load Balancer, dan harus merupakan penyeimbang beban internal.
- Penyeimbang beban dan subnet harus dimiliki oleh virtual private cloud (VPC) yang sama.
- Penyeimbang beban HTTPS dapat menggunakan sertifikat TLS yang ditandatangani sendiri atau publik.
- Anda harus memberikan nama domain untuk aplikasi Anda. Ini adalah nama DNS publik yang akan digunakan pengguna Anda untuk mengakses aplikasi Anda. Anda juga perlu memberikan sertifikat SSL publik dengan CN yang cocok dengan nama domain ini. Anda dapat membuat atau mengimpor sertifikat menggunakanAWS Certificate Manager.

Untuk membuat titik akhir penyeimbang beban

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih titik akhir Akses Terverifikasi.
3. Pilih Buat titik akhir Akses Terverifikasi.
4. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk titik akhir.
5. Untuk grup Akses Terverifikasi, pilih grup Akses Terverifikasi untuk titik akhir.
6. Untuk detail Aplikasi, lakukan hal berikut:

    a. Untuk domain Aplikasi, masukkan nama DNS untuk aplikasi Anda.

b. Di bawah Sertifikat domain ARN, pilih sertifikat TLS publik.

7. Untuk detail Endpoint, lakukan hal berikut:

   a. Untuk jenis Lampiran, pilih VPC.

   b. Untuk grup Keamanan, pilih grup keamanan untuk titik akhir. Lalu lintas dari titik akhir Akses Terverifikasi yang memasuki penyeimbang beban Anda akan dikaitkan dengan grup keamanan ini.

   c. Untuk awalan domain Endpoint, masukkan pengenal kustom untuk menambahkan nama DNS yang dihasilkan Akses Terverifikasi untuk titik akhir.

   d. Untuk tipe Endpoint, pilih Load balancer.

   e. Untuk Protokol, pilih HTTPS atau HTTP.

   f. Di bawah Port, masukkan nomor port.

   g. Untuk Load balancer ARN, pilih load balancer.

   h. Untuk Subnet, pilih subnet untuk penyeimbang beban Anda.

8. (Opsional) Untuk definisi Kebijakan, masukkan kebijakan Akses Terverifikasi untuk titik akhir.

9. (Opsional) Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag.

10. Pilih Buat titik akhir Akses Terverifikasi.

Membuat titik akhir antarmuka jaringan untuk Akses Terverifikasi

Gunakan prosedur berikut untuk membuat titik akhir antarmuka jaringan.

Persyaratan

- Hanya lalu lintas IPv4 yang didukung.
- Hanya protokol HTTP dan HTTPS yang didukung.
- Antarmuka jaringan harus termasuk dalam virtual private cloud (VPC) yang sama dengan grup keamanan.
- Kami menggunakan IP pribadi pada antarmuka jaringan untuk meneruskan lalu lintas.
- Anda harus memberikan nama domain untuk aplikasi Anda. Ini adalah nama DNS publik yang akan digunakan pengguna Anda untuk mengakses aplikasi Anda. Anda juga perlu memberikan sertifikat SSL publik dengan CN yang cocok dengan nama domain ini. Anda dapat membuat atau mengimpor sertifikat menggunakanAWS Certificate Manager.

Untuk membuat endpoint antarmuka jaringan

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih titik akhir Akses Terverifikasi.
3. Pilih Buat titik akhir Akses Terverifikasi.
4. (Opsional) Untuk tag Nama dan Deskripsi, masukkan nama dan deskripsi untuk titik akhir.
5. Untuk grup Akses Terverifikasi, pilih grup Akses Terverifikasi untuk titik akhir.
6. Untuk detail Aplikasi, lakukan hal berikut:
   a. Untuk domain Aplikasi, masukkan nama DNS untuk aplikasi Anda.
   b. Di bawah Sertifikat domain ARN, pilih sertifikat TLS publik.
7. Untuk detail Endpoint, lakukan hal berikut:
   a. Untuk jenis Lampiran, pilih VPC.
   b. Untuk grup Keamanan, pilih grup keamanan untuk titik akhir. Lalu lintas dari titik akhir Akses Terverifikasi yang memasuki antarmuka jaringan Anda akan dikaitkan dengan grup keamanan ini.
   c. Untuk awalan domain Endpoint, masukkan pengenal kustom untuk menambahkan nama DNS yang dihasilkan Akses Terverifikasi untuk titik akhir.
   d. Untuk tipe Endpoint, pilih Network interface.
   e. Untuk Protokol, pilih HTTPS atau HTTP.
   f. Di bawah Port, masukkan nomor port.
   g. Untuk antarmuka Jaringan, pilih antarmuka jaringan.
8. (Opsional) Untuk definisi Kebijakan, masukkan kebijakan Akses Terverifikasi untuk titik akhir.
9. (Opsional) Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag.
10. Pilih Buat titik akhir Akses Terverifikasi.

Izinkan lalu lintas yang berasal dari titik akhir Akses Terverifikasi Anda

Anda dapat mengonfigurasi grup keamanan untuk aplikasi Anda sehingga memungkinkan lalu lintas yang berasal dari titik akhir Akses Terverifikasi Anda. Anda melakukannya dengan

menambahkan aturan masuk yang menentukan grup keamanan untuk titik akhir sebagai sumber. Kami menyarankan Anda menghapus aturan masuk tambahan, sehingga aplikasi Anda hanya menerima lalu lintas dari titik akhir Akses Terverifikasi Anda.

Kami menyarankan Anda untuk mempertahankan aturan keluar yang ada.

Untuk memperbarui aturan grup keamanan untuk aplikasi Anda

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih titik akhir Akses Terverifikasi.
3. Pilih titik akhir Akses Terverifikasi, temukan ID grup Keamanan di tab Detail, dan salin ID grup keamanan untuk titik akhir Anda.
4. Di panel navigasi, pilih Grup keamanan.
5. Pilih kotak centang untuk grup keamanan yang terkait dengan target Anda, lalu pilih Tindakan, Edit aturan masuk.
6. Untuk menambahkan aturan grup keamanan yang mengizinkan lalu lintas yang berasal dari titik akhir Akses Terverifikasi, lakukan hal berikut:
   a. Pilih Add rule (Tambahkan aturan).
   b. Untuk Jenis, pilih Semua lalu lintas atau lalu lintas tertentu yang akan diizinkan.
   c. Untuk Sumber, pilih Kustom dan tempel ID grup keamanan untuk titik akhir Anda.
7. (Opsional) Untuk mewajibkan lalu lintas hanya berasal dari titik akhir Akses Terverifikasi Anda, hapus aturan grup keamanan masuk lainnya.
8. Pilih Save rules (Simpan aturan).

Ubah titik akhir Akses Terverifikasi

Setelah Anda membuat titik akhir Akses Terverifikasi, Anda dapat memperbarui konfigurasinya.

Untuk mengubah titik akhir Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih titik akhir Akses Terverifikasi.
3. Pilih titik akhir.
4. Pilih Tindakan, Ubah titik akhir Akses Terverifikasi.
5. Ubah detail titik akhir sesuai kebutuhan.

6. Pilih Ubah titik akhir Akses Terverifikasi.

Ubah kebijakan titik akhir Akses Terverifikasi

Setelah membuat titik akhir Akses Terverifikasi, Anda dapat mengubah kebijakannya.

Untuk mengubah kebijakan titik akhir Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih titik akhir Akses Terverifikasi.
3. Pilih titik akhir yang kebijakannya ingin Anda ubah.
4. Pilih Tindakan, Ubah kebijakan titik akhir Akses Terverifikasi.
5. (Opsional) Aktifkan atau nonaktifkan Aktifkan kebijakan tergantung pada tujuan Anda saat ini.
6. (Opsional) Untuk Kebijakan, masukkan kebijakan Akses Terverifikasi untuk diterapkan pada titik akhir.
7. Pilih Ubah kebijakan titik akhir Akses Terverifikasi.

Menghapus titik akhir Akses Terverifikasi

Setelah selesai dengan titik akhir Akses Terverifikasi, Anda dapat menghapusnya.

Untuk menghapus titik akhir Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih titik akhir Akses Terverifikasi.
3. Pilih titik akhir.
4. Pilih Tindakan, Hapus titik akhir Akses Terverifikasi.
5. Ketika diminta konfirmasi, masukkan **`delete`** lalu pilih Hapus.

Data kepercayaan dari penyedia kepercayaan

Data kepercayaan adalah data yang dikirim ke Akses AWS Terverifikasi dari penyedia kepercayaan. Kadang-kadang disebut sebagai “klaim pengguna” atau “konteks kepercayaan” juga. Data umumnya mencakup informasi tentang pengguna atau perangkat. Contoh data kepercayaan termasuk email pengguna, keanggotaan grup, versi sistem operasi perangkat, status keamanan perangkat, dan banyak lagi. Informasi yang dikirim bervariasi berdasarkan penyedia kepercayaan, jadi Anda harus merujuk ke dokumentasi penyedia kepercayaan Anda untuk daftar data kepercayaan yang lengkap dan diperbarui.

Namun, dengan menggunakan kemampuan pencatatan Akses Terverifikasi, Anda juga dapat melihat data kepercayaan apa yang dikirim dari penyedia kepercayaan Anda. Ini bisa sangat berguna saat mendefinisikan kebijakan yang mengizinkan atau menolak akses ke aplikasi Anda. Untuk informasi tentang menyertakan konteks kepercayaan di log Anda, lihatTermasuk konteks kepercayaan.

Bagian ini berisi contoh data kepercayaan dan contoh untuk memulai penulisan kebijakan. Informasi yang diberikan di sini dimaksudkan untuk tujuan ilustrasi saja dan bukan sebagai referensi resmi.

Daftar Isi

Konteks default Akses Terverifikasi

AWSAkses Terverifikasi mencakup beberapa elemen tentang permintaan HTTP saat ini secara default di semua evaluasi Cedar terlepas dari penyedia kepercayaan Anda yang dikonfigurasi. Saat kebijakan dievaluasi, Akses Terverifikasi menyertakan data tentang permintaan HTTP saat ini dalam konteks Cedar di bawah. `context.http_request key` Anda dapat menulis kebijakan yang mengevaluasi data jika Anda memilih. Skema JSON berikut menunjukkan data mana yang termasuk dalam evaluasi.

```
{
     "title": "HTTP Request data included by Verified Access",
     "type": "object",
     "properties": {
```

```
        "user_agent": {
            "type": "string",
            "description": "The value of the User-Agent request header"
        },
        "x_forwarded_for": {
            "type": "string",
            "description": "The value of the X-Forwarded-For request header"
        },
        "http_method": {
            "type": "string",
            "description": "The HTTP Method provided (e.g. GET or POST)"
        },
        "hostname": {
            "type": "string",
            "description": "The value of the Host request header"
        },
        "port": {
           "type": "integer",
           "description": "The value of the verified access endpoint port"
        },
        "client_ip": {
            "type": "string",
            "description": "User ip connecting to the verified access endpoint"
        }
    }
}
```

Berikut ini adalah contoh kebijakan yang mengevaluasi terhadap data permintaan HTTP.

```
forbid(principal, action, resource) when {
   context.http_request.http_method == "POST"
   && !(context.identity.roles.contains("Administrator"))
};
```

AWSPusat Identitas IAM

Ketika kebijakan dievaluasi, jika Anda mendefinisikan AWS IAM Identity Center sebagai penyedia kepercayaan, Akses AWS Terverifikasi menyertakan data kepercayaan dalam konteks Cedar di bawah kunci yang Anda tentukan sebagai “Nama Referensi Kebijakan” pada konfigurasi penyedia kepercayaan. Anda dapat menulis kebijakan yang mengevaluasi terhadap data kepercayaan jika Anda memilih.

> **Note**
>
> Kunci konteks untuk penyedia kepercayaan Anda berasal dari nama referensi kebijakan yang Anda konfigurasikan saat membuat penyedia kepercayaan. Misalnya, jika Anda mengonfigurasi nama referensi kebijakan sebagai “idp123", kunci konteksnya adalah “context.idp123". Periksa apakah Anda menggunakan kunci konteks yang benar saat membuat kebijakan.

Skema JSON berikut menunjukkan data mana yang termasuk dalam evaluasi.

```
{
   "title": "AWS IAM Identity Center context specification",
   "type": "object",
   "properties": {
     "user": {
       "type": "object",
       "properties": {
         "user_id": {
           "type": "string",
           "description": "a unique user id generated by AWS IdC"
         },
         "user_name": {
           "type": "string",
           "description": "username provided in the directory"
         },
         "email": {
           "type": "object",
           "properties": {
             "address": {
               "type": "email",
               "description": "email address associated with the user"
             },
             "verified": {
               "type": "boolean",
               "description": "whether the email address has been verified by AWS IdC"
             }
           }
         }
       }
     },
     "groups": {
```

```
       "type": "object",
       "description": "A list of groups the user is a member of",
       "patternProperties": {
         "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{12}$": {
           "type": "object",
           "description": "The Group ID of the group",
           "properties": {
             "group_name": {
               "type": "string",
               "description": "The customer-provided name of the group"
             }
           }
         }
       }
     }
   }
 }
```

Berikut ini adalah contoh kebijakan yang mengevaluasi terhadap data kepercayaan yang diberikan oleh AWS IAM Identity Center.

```
permit(principal, action, resource) when {
   context.idc.user.email.verified == true
   // User is in the "sales" group with specific ID
   && context.idc.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
 };
```

> **Note**
>
> Karena nama grup dapat diubah, IAM Identity Center mengacu pada grup yang menggunakan ID grup mereka. Ini membantu menghindari melanggar pernyataan kebijakan saat mengubah nama grup.

Penyedia kepercayaan pihak ketiga

Bagian ini menjelaskan data kepercayaan yang diberikan kepada Akses AWS Terverifikasi oleh penyedia kepercayaan pihak ketiga.

> **Note**
>
> Kunci konteks untuk penyedia kepercayaan Anda berasal dari nama referensi kebijakan yang Anda konfigurasikan saat membuat penyedia kepercayaan. Misalnya, jika Anda mengonfigurasi nama referensi kebijakan sebagai “idp123", kunci konteksnya adalah “context.idp123". Pastikan Anda menggunakan kunci konteks yang benar saat membuat kebijakan.

Daftar Isi

Ekstensi browser

Jika Anda berencana untuk memasukkan konteks kepercayaan perangkat ke dalam kebijakan akses Anda, maka Anda akan memerlukan ekstensi browser Akses AWS Terverifikasi, atau ekstensi browser mitra lain. Akses Terverifikasi saat ini mendukung browser Google Chrome dan Mozilla Firefox.

Saat ini kami mendukung tiga penyedia kepercayaan perangkat: Jamf (yang mendukung perangkat macOS) CrowdStrike , (yang mendukung perangkat Windows 11 dan Windows 10), JumpCloud dan (yang mendukung Windows dan macOS).

- Jika Anda menggunakan data kepercayaan Jamf dalam kebijakan Anda, pengguna Anda harus mengunduh dan menginstal ekstensi browser Akses AWS Terverifikasi dari toko web Chrome atau situs Add-on Firefox di perangkat mereka.
- Jika Anda menggunakan data CrowdStrikekepercayaan dalam kebijakan Anda, pertama-tama pengguna Anda harus menginstal Host Pesan Asli Akses AWS Terverifikasi (tautan unduhan langsung). Komponen ini diperlukan untuk mendapatkan data kepercayaan dari CrowdStrike agen yang berjalan di perangkat pengguna. Kemudian, setelah menginstal komponen ini, pengguna harus menginstal ekstensi browser Akses AWS Terverifikasi dari toko web Chrome atau situs Add-on Firefox di perangkat mereka.

- Jika Anda menggunakan JumpCloud, pengguna Anda harus memiliki ekstensi JumpCloud browser dari toko web Chrome atau situs Add-on Firefox yang diinstal pada perangkat mereka.

Jamf

Jamf adalah penyedia kepercayaan pihak ketiga. Ketika kebijakan dievaluasi, jika Anda mendefinisikan Jamf sebagai penyedia kepercayaan, Akses Terverifikasi menyertakan data kepercayaan dalam konteks Cedar di bawah kunci yang Anda tentukan sebagai “Nama Referensi Kebijakan” pada konfigurasi penyedia kepercayaan. Anda dapat menulis kebijakan yang mengevaluasi terhadap data kepercayaan jika Anda memilih. Skema JSON berikut menunjukkan data mana yang termasuk dalam evaluasi.

Untuk informasi selengkapnya tentang penggunaan Jamf dengan Akses AWS Terverifikasi, lihat Mengintegrasikan AWS Verified Access dengan Jamf Device Identity di situs web Jamf.

```
{
    "title": "Jamf device data specification",
    "type": "object",
    "properties": {
        "iss": {
            "type": "string",
            "description": "\"Issuer\" - the Jamf customer ID"
        },
        "iat": {
            "type": "integer",
            "description": "\"Issued at Time\" - a unixtime (seconds since epoch) value
 of when the device information data was generated"
        },
        "exp": {
            "type": "integer",
            "description": "\"Expiration\" - a unixtime (seconds since epoch) value for
 when this device information is no longer valid"
        },
        "sub": {
            "type": "string",
            "description": "\"Subject\" - either the hardware UID or a value generated
 based on device location"
        },
        "groups": {
            "type": "array",
            "description": "Group IDs from UEM connector sync",
```

```
            "items": {
                "type": "string"
            }
        },
        "risk": {
            "type": "string",
            "enum": [
                "HIGH",
                "MEDIUM",
                "LOW",
                "SECURE",
                "NOT_APPLICABLE"
            ],
            "description": "a Jamf-reported level of risk associated with the device."
        },
        "osv": {
            "type": "string",
            "description": "The version of the OS that is currently running, in Apple
 version number format (https://support.apple.com/en-us/HT201260)"
        }
    }
}
```

Berikut ini adalah contoh kebijakan yang mengevaluasi terhadap data kepercayaan yang diberikan oleh Jamf.

```
permit(principal, action, resource) when {
   context.jamf.risk == "LOW"
};
```

Cedar menyediakan `.contains()` fungsi yang berguna untuk membantu dengan enum seperti skor risiko Jamf.

```
permit(principal, action, resource) when {
   ["LOW", "SECURE"].contains(context.jamf.risk)
};
```

CrowdStrike

CrowdStrike adalah penyedia kepercayaan pihak ketiga. Ketika kebijakan dievaluasi, jika Anda mendefinisikan CrowdStrike sebagai penyedia kepercayaan, Akses Terverifikasi menyertakan

data kepercayaan dalam konteks Cedar di bawah kunci yang Anda tentukan sebagai “Nama Referensi Kebijakan” pada konfigurasi penyedia kepercayaan. Anda dapat menulis kebijakan yang mengevaluasi terhadap data kepercayaan jika Anda memilih. Skema JSON berikut menunjukkan data mana yang termasuk dalam evaluasi.

Untuk informasi selengkapnya tentang penggunaan CrowdStrike dengan Akses AWS Terverifikasi, lihat Mengamankan aplikasi pribadi dengan CrowdStrike dan Akses AWS Terverifikasi di GitHub situs web.

```
{
  "title": "CrowdStrike device data specification",
  "type": "object",
  "properties": {
    "assessment": {
      "type": "object",
      "description": "Data about CrowdStrike's assessment of the device",
      "properties": {
        "overall": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts as a weighted
 average of the OS and and Sensor Config scores"
        },
        "os": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the OS-
specific settings monitored on the host"
        },
        "sensor_config": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the
 different sensor policies monitored on the host"
        },
        "version": {
          "type": "string",
          "description": "The version of the scoring algorithm being used"
        }
      }
    },
    "cid": {
      "type": "string",
      "description": "Customer ID (CID) unique to the customer's environemnt"
    },
    "exp": {
```

```
      "type": "integer",
      "description": "unixtime, The expiration time of the token"
    },
    "iat": {
      "type": "integer",
      "description": "unixtime, The issued time of the token"
    },
    "jwk_url": {
      "type": "string",
      "description": "URL that details the JWT signing"
    },
    "platform": {
      "type": "string",
      "enum": ["Windows 10", "Windows 11", "macOS"],
      "description": "Operating system of the endpoint"
    },
    "serial_number": {
      "type": "string",
      "description": "The serial number of the device derived by unique system
 information"
    },
    "sub": {
      "type": "string",
      "description": "Unique CrowdStrike Agent ID (AID) of machine"
    },
    "typ": {
      "type": "string",
      "enum": ["crowdstrike-zta+jwt"],
      "description": "Generic name for this JWT media. Client MUST reject any other
 type"
    }
  }
}
```

Berikut ini adalah contoh kebijakan yang mengevaluasi terhadap data kepercayaan yang diberikan oleh CrowdStrike.

```
permit(principal, action, resource) when {
   context.crowdstrike.assessment.overall > 50
};
```

JumpCloud

JumpCloud adalah penyedia kepercayaan pihak ketiga. Ketika kebijakan dievaluasi, jika Anda mendefinisikan JumpCloud sebagai penyedia kepercayaan, Akses Terverifikasi menyertakan data kepercayaan dalam konteks Cedar di bawah kunci yang Anda tentukan sebagai “Nama Referensi Kebijakan” pada konfigurasi penyedia kepercayaan. Anda dapat menulis kebijakan yang mengevaluasi terhadap data kepercayaan jika Anda memilih. Skema JSON berikut menunjukkan data mana yang termasuk dalam evaluasi.

Untuk informasi selengkapnya tentang penggunaan JumpCloud dengan Akses AWS Terverifikasi, lihat Mengintegrasikan JumpCloud dan Akses AWS Terverifikasi di JumpCloud situs web.

```
{
  "title": "JumpCloud device data specification",
  "type": "object",
  "properties": {
    "device": {
      "type": "object",
      "description": "Properties of the device",
      "properties": {
        "is_managed": {
          "type": "boolean",
          "description": "Boolean to indicate if the device is under management"
        }
      }
    },
    "exp": {
      "type": "integer",
      "description": "Expiration. Unixtime of the token's expiration."
    },
    "durt_id": {
      "type": "string",
      "description": "Device User Refresh Token ID. Unique ID that represents the
 device + user."
    },
    "iat": {
      "type": "integer",
      "description": "Issued At. Unixtime of the token's issuance."
    },
    "iss": {
      "type": "string",
      "description": "Issuer. This will be 'go.jumpcloud.com'"
```

```
    },
    "org_id": {
      "type": "string",
      "description": "The JumpCloud Organization ID"
    },
    "sub": {
      "type": "string",
      "description": "Subject. The managed JumpCloud user ID on the device."
    },
    "system": {
      "type": "string",
      "description": "The JumpCloud system ID"
    }
  }
}
```

Berikut ini adalah contoh kebijakan yang mengevaluasi terhadap konteks kepercayaan yang diberikan oleh JumpCloud.

```
permit(principal, action, resource) when {
   context.jumpcloud.org_id = 'Unique_orgnaization_identifier'
};
```

Klaim pengguna lulus dan verifikasi tanda tangan

Setelah instance Akses AWS Terverifikasi berhasil mengautentikasi pengguna, instans mengirimkan klaim pengguna yang diterima dari iDP ke titik akhir Akses Terverifikasi. Klaim pengguna ditandatangani sehingga aplikasi dapat memverifikasi tanda tangan dan klaim dikirim oleh Akses Terverifikasi. Selama proses ini, header HTTP berikut ditambahkan:

`x-amzn-ava-user-context`

Header ini berisi klaim pengguna dalam format token web JSON (JWT). Format JWT mencakup header, payload, dan tanda tangan yang dikodekan URL base64. Akses Terverifikasi menggunakan ES384 (algoritma tanda tangan ECDSA menggunakan algoritma hash SHA-384) untuk menghasilkan tanda tangan JWT.

Aplikasi dapat menggunakan klaim ini untuk personalisasi atau pengalaman khusus pengguna lainnya. Pengembang aplikasi harus mendidik diri mereka sendiri mengenai tingkat keunikan dan verifikasi setiap klaim yang diberikan oleh penyedia identitas sebelum digunakan. Secara umum, `sub` klaim adalah cara terbaik untuk mengidentifikasi pengguna tertentu.

Daftar Isi

Contoh: Menandatangani JWT untuk klaim pengguna OIDC

Contoh berikut menunjukkan seperti apa header dan payload untuk klaim pengguna OIDC dalam format JWT.

Contoh header:

```
{
"alg": "ES384",
"kid": "12345678-1234-1234-1234-123456789012",
"signer": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/vai-
abc123xzy321a2b3c",
"iss": "OIDC Issuer URL"
"exp": "expiration" (120 secs)
}
```

Contoh muatan:

```
{
"sub": "xyzsubject",
"email": "xxx@amazon.com",
"email_verified": true,
"groups": [
"Engineering",
"finance"
]
}
```

Contoh: Menandatangani JWT untuk klaim pengguna IAM Identity Center

Contoh berikut menunjukkan seperti apa header dan payload untuk klaim pengguna IAM Identity Center dalam format JWT.

> **Note**
>
> Untuk IAM Identity Center, hanya informasi pengguna yang akan dimasukkan dalam klaim.

Contoh header:

```
{
"alg": "ES384",
"kid": "12345678-1234-1234-1234-123456789012",
"signer": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/vai-
abc123xzy321a2b3c",
"iss": "arn:aws:ec2:us-east-1:123456789012:verified-access-trust-provider/vatp-
abc123xzy321a2b3c",
"exp": "expiration" (120 secs)
}
```

Contoh muatan:

```
{
"user": {
        "user_id": "f478d4c8-a001-7064-6ea6-12423523",
        "user_name": "test-123",
        "email": {
            "address": "test@amazon.com",
            "verified": false
        }
    }
}
```

Kunci publik

Karena instans Akses Terverifikasi tidak mengenkripsi klaim pengguna, sebaiknya Anda mengonfigurasi titik akhir Akses Terverifikasi untuk menggunakan HTTPS. Jika Anda mengonfigurasi titik akhir Akses Terverifikasi untuk menggunakan HTTP, pastikan untuk membatasi lalu lintas ke titik akhir menggunakan grup keamanan.

Kami menyarankan Anda memverifikasi tanda tangan sebelum melakukan otorisasi berdasarkan klaim. Untuk mendapatkan kunci publik, dapatkan ID kunci dari header JWT dan gunakan untuk mencari kunci publik dari titik akhir. Titik akhir untuk masing-masing Wilayah AWS adalah sebagai berikut:

`https://public-keys.prod.verified-access.<region>.amazonaws.com/<key-id>`

Contoh: Mengambil dan mendekode JWT

Contoh kode berikut menunjukkan cara mendapatkan ID kunci, kunci publik, dan payload di Python 3.9.

```
import jwt
import requests
import base64
import json

# Step 1: Get the key id from JWT headers (the kid field)
encoded_jwt = headers.dict['x-amzn-ava-user-context']
jwt_headers = encoded_jwt.split('.')[0]
decoded_jwt_headers = base64.b64decode(jwt_headers)
decoded_jwt_headers = decoded_jwt_headers.decode("utf-8")
decoded_json = json.loads(decoded_jwt_headers)
kid = decoded_json['kid']

# Step 2: Get the public key from Regional endpoint
url = 'https://public-keys.prod.verified-access.' + region + '.amazonaws.com/' + kid
req = requests.get(url)
pub_key = req.text

# Step 3: Get the payload
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES384'])
```

Kebijakan Akses Terverifikasi

AWSKebijakan Akses Terverifikasi memungkinkan Anda menentukan aturan untuk mengakses aplikasi yang dihosting. AWS Mereka ditulis dalam Cedar, bahasa AWS kebijakan. Menggunakan Cedar, Anda dapat membuat kebijakan yang dievaluasi terhadap konteks kepercayaan yang dikirim dari identitas atau penyedia kepercayaan berbasis perangkat yang Anda konfigurasikan untuk digunakan dengan Akses Terverifikasi.

Untuk informasi lebih rinci tentang bahasa kebijakan Cedar, lihat Panduan Referensi Cedar.

Bagian ini menjelaskan bagaimana kebijakan Akses Terverifikasi terstruktur, isinya, cara mendefinisikannya, dan memberikan beberapa contoh.

Daftar Isi

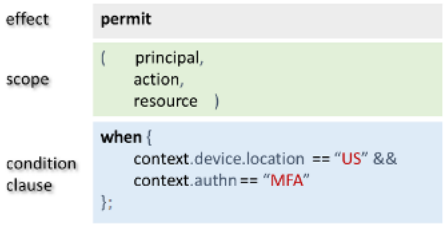
Bekerja dengan kebijakan untuk Akses Terverifikasi

Saat membuat grup Akses Terverifikasi atau membuat titik akhir Akses Terverifikasi, Anda memiliki opsi untuk menentukan kebijakan Akses Terverifikasi. Anda dapat membuat grup atau titik akhir tanpa menentukan kebijakan Akses Terverifikasi, tetapi semua permintaan akses akan diblokir hingga Anda menentukan kebijakan.

Untuk menambah atau mengubah kebijakan pada grup Akses Terverifikasi atau titik akhir yang ada setelah dibuat, lihat Memodifikasi kebijakan grup Akses Terverifikasi atauUbah kebijakan titik akhir Akses Terverifikasi.

Struktur pernyataan kebijakan

Bagian ini menjelaskan pernyataan kebijakan Akses AWS Terverifikasi dan bagaimana hal itu dievaluasi. Anda dapat memiliki beberapa pernyataan dalam satu kebijakan Akses Terverifikasi. Diagram berikut menunjukkan struktur kebijakan Akses Terverifikasi.



Kebijakan ini berisi bagian-bagian berikut:

- Efek - Menentukan apakah pernyataan kebijakan adalah `permit` (`Allow`) atau `forbid` (`Deny`).
- Lingkup - Menentukan prinsip, tindakan, dan sumber daya yang efeknya berlaku. Anda dapat membiarkan ruang lingkup di Cedar tidak terdefinisi dengan tidak mengidentifikasi prinsip, tindakan, atau sumber daya tertentu (seperti yang ditunjukkan pada contoh sebelumnya). Dalam hal ini, kebijakan berlaku untuk semua prinsip, tindakan, dan sumber daya yang mungkin.
- Kondisi klausa - Menentukan konteks di mana efek berlaku.

> **Important**
>
> Untuk Akses Terverifikasi, kebijakan diungkapkan sepenuhnya dengan mengacu pada konteks kepercayaan dalam klausul kondisi. Ruang lingkup kebijakan harus selalu tetap tidak terdefinisi. Anda kemudian dapat menentukan akses menggunakan identitas dan konteks kepercayaan perangkat dalam klausa kondisi.

Contoh kebijakan sederhana

```
permit(principal,action,resource)
when{
 context.<policy-reference-name>.<attribute> &&
 context.<policy-reference-name>.<attribute2>
};
```

Pada contoh sebelumnya, perhatikan bahwa Anda dapat menggunakan lebih dari satu klausa kondisi dalam pernyataan kebijakan menggunakan operator. && Bahasa kebijakan Cedar memberi Anda kekuatan ekspresif untuk membuat pernyataan kebijakan yang bersifat adat, berbutir halus, dan ekstensif. Untuk contoh tambahan, lihat Contoh kebijakan .

Evaluasi kebijakan

Dokumen kebijakan adalah satu set dari satu atau lebih pernyataan kebijakan (`permit`atau `forbid` pernyataan). Kebijakan ini berlaku jika klausa kondisional (`when`pernyataan) benar. Agar dokumen kebijakan memungkinkan akses, setidaknya satu kebijakan izin dalam dokumen harus berlaku dan tidak ada kebijakan larangan yang dapat diterapkan. Jika tidak ada kebijakan izin yang berlaku dan/atau satu atau lebih kebijakan larangan berlaku, maka dokumen kebijakan tersebut menolak akses. Jika Anda telah menetapkan dokumen kebijakan untuk grup Akses Terverifikasi dan titik akhir Akses Terverifikasi, kedua dokumen harus mengizinkan akses. Jika Anda belum menetapkan dokumen kebijakan untuk titik akhir Akses Terverifikasi, hanya kebijakan grup Akses Terverifikasi yang perlu diakses.

> **Note**
>
> AWSAkses Terverifikasi memvalidasi sintaks saat Anda membuat kebijakan, tetapi tidak memvalidasi data yang Anda masukkan ke dalam klausa bersyarat.

Operator bawaan

Saat membuat konteks kebijakan Akses AWS Terverifikasi menggunakan berbagai kondisi, seperti yang dibahas dalamStruktur pernyataan kebijakan, Anda dapat menggunakan && operator untuk menambahkan kondisi tambahan. Ada juga banyak operator bawaan lainnya yang dapat Anda gunakan untuk menambahkan kekuatan ekspresif tambahan pada kondisi kebijakan Anda. Tabel berikut berisi semua operator bawaan untuk referensi.

Operator	Jenis dan kelebihan beban	Deskripsi
!	Boolean → Boolean	Logis tidak.
==	apa saja → apa saja	Kesetaraan. Bekerja pada argumen jenis apa pun,

Operator	Jenis dan kelebihan beban	Deskripsi
		bahkan jika tipenya tidak cocok. Nilai dari berbagai jenis tidak pernah sama satu sama lain.
!=	apa saja → apa saja	Ketimpangan; kebalikan dari kesetaraan (lihat di atas).
<	(panjang, panjang) → Boolean	Bilangan bulat panjang kurang dari.
<=	(panjang, panjang) → Boolean	Bilangan bulat panjang less-than-or-equal -ke.
>	(panjang, panjang) → Boolean	Bilangan bulat panjang lebih besar dari.
>=	(panjang, panjang) → Boolean	Bilangan bulat panjang greater-than-or-equal -ke.
in	(entitas, entitas) → Boolean	Keanggotaan hierarki (refleksi f: A dalam A selalu benar).
	(entitas, set (entitas)) → Boolean	Keanggotaan hierarki: A di [B, C,...] benar jika (A dan B) || (A dalam C) ||... kesalahan jika himpunan berisi non-entitas.
&&	(Boolean, Boolean) → Boolean	Logis dan (hubungan arus pendek).
||	(Boolean, Boolean) → Boolean	Logis atau (hubungan arus pendek).
.ada ()	entitas → Boolean	Keberadaan entitas.

Operator	Jenis dan kelebihan beban	Deskripsi
memiliki	(entitas, atribut) → Boolean	Operator infix. `e has f`menguji apakah catatan atau entitas `e` memiliki pengikata n untuk atribut`f`. Mengembal ikan `false` jika `e` tidak ada atau jika `e` memang ada tetapi tidak memiliki atribut`f`. Atribut dapat dinyatakan sebagai pengidentifikasi atau string literal.
suka	(string, string) → Boolean	Operator infix. `t like p`memeriksa apakah teks `t` cocok dengan polap, yang mungkin termasuk karakter wildcard `*` yang cocok dengan 0 atau lebih dari karakter apa pun. Untuk mencocokkan karakter bintang literal`t`, Anda dapat menggunakan urutan karakter lolos khusus `\*` dip.
.berisi ()	(set, apa saja) → Boolean	Tetapkan keanggotaan (adalah B elemen A).
.containsAll ()	(set, atur) → Boolean	Tes jika set A berisi semua elemen dalam himpunan B.
.containsAny ()	(set, atur) → Boolean	Tes jika set A berisi salah satu elemen dalam himpunan B.

Komentar kebijakan

Anda dapat menyertakan pernyataan komentar dalam kebijakan Akses AWS Terverifikasi Anda. Komentar didefinisikan sebagai baris yang dimulai dengan `//` dan diakhiri dengan baris baru.

Contoh berikut menunjukkan pernyataan komentar dalam kebijakan.

```
// this policy grants access to users in a given domain with trusted devices
permit(principal, action, resource)
when {
  // the user's email address is in the @example.com domain
  context.idc.user.email.address.contains("@example.com")
  // Jamf thinks the user's computer is low risk or secure.
  && ["LOW", "SECURE"].contains(context.jamf.risk)
};
```

Logika kebijakan hubung singkat

Anda mungkin ingin menulis kebijakan Akses AWS Terverifikasi yang mengevaluasi data yang mungkin atau mungkin tidak ada dalam konteks tertentu. Jika Anda mereferensikan data dalam konteks yang tidak ada, Cedar akan menghasilkan kesalahan dan mengevaluasi kebijakan untuk menolak akses, terlepas dari maksud Anda. Misalnya, ini akan menghasilkan penolakan, karena `fake_provider` dan `bogus_key` tidak ada dalam konteks ini.

```
permit(principal, action, resource) when {
  context.fake_provider.bogus_key > 42
};
```

Untuk menghindari situasi ini, Anda dapat memeriksa untuk melihat apakah ada kunci menggunakan `has` operator. Jika `has` operator mengembalikan false, evaluasi lebih lanjut dari pernyataan berantai berhenti, dan Cedar tidak menghasilkan kesalahan saat mencoba mereferensikan item yang tidak ada.

```
permit(principal, action, resource) when {
  context.identity.user has "some_key" && context.identity.user.some_key > 42
};
```

Ini sangat berguna ketika menentukan kebijakan yang mereferensikan dua penyedia kepercayaan yang berbeda.

```
permit(principal, action, resource) when {
  // user is in an allowed group
  context.aws_idc.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
  &&(
    (
```

```
      // if CrowdStrike data is present,
      // permit if CrowdStrike's overall assessment is over 50
      context has "crowdstrike" && context.crowdstrike.assessment.overall > 50
    )
    ||
    (
      // if Jamf data is present,
      // permit if Jamf's risk score is acceptable
      context has "jamf" && ["LOW", "NOT_APPLICABLE", "MEDIUM",
 "SECURE"].contains(context.jamf.risk)
    )
  )
};
```

Contoh kebijakan

Contoh 1: Membuat kebijakan untuk IAM Identity Center

> **Note**
>
> Karena nama grup dapat diubah, IAM Identity Center mengacu pada grup yang menggunakan ID grup mereka. Ini membantu menghindari melanggar pernyataan kebijakan saat mengubah nama grup.

Contoh kebijakan berikut memungkinkan akses hanya ketika pengguna milik `finance` grup (yang memiliki ID grup`c242c5b0-6081-1845-6fa8-6e0d9513c107`) dan memiliki alamat email terverifikasi.

```
permit(principal,action,resource)
when {
    context.<policy-reference-name>.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
    && context.<policy-reference-name>.user.email.verified == true
};
```

Contoh 1b: Menambahkan lebih banyak kondisi ke pernyataan kebijakan untuk IAM Identity Center

Contoh kebijakan berikut mengizinkan akses hanya ketika pengguna termasuk dalam `finance` grup (yang memiliki ID grup`c242c5b0-6081-1845-6fa8-6e0d9513c107`), memiliki alamat email terverifikasi, dan skor risiko perangkat Jamf adalah`L0W`.

```
permit(principal,action,resource)
when {
    context.<policy-reference-name>.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
    && context.<policy-reference-name>.user.email.verified == true
    && context.jamf.risk == "LOW"
};
```

Contoh 2: Kebijakan yang sama untuk penyedia OIDC pihak ke-3

Contoh kebijakan berikut memungkinkan akses hanya ketika pengguna berasal dari grup “keuangan”, mereka memiliki alamat email terverifikasi, dan skor risiko perangkat Jamf RENDAH.

```
permit(principal,action,resource)
when {
     context.<policy-reference-name>.groups.contains("finance")
     && context.<policy-reference-name>.email_verified == true
     && context.jamf.risk == "LOW"
};
```

Contoh 3: Menggunakan CrowdStrike

Contoh kebijakan berikut memungkinkan akses ketika skor penilaian keseluruhan lebih besar dari 50.

```
permit(principal,action,resource)
when {
    context.crwd.assessment.overall > 50
};
```

Contoh 4: Bekerja dengan karakter khusus

Contoh berikut menunjukkan cara menulis kebijakan jika properti context menggunakan : (titik koma), yang merupakan karakter cadangan dalam bahasa kebijakan.

```
permit(principal, action, resource)
when {
    context.<policy-reference-name>["namespace:groups"].contains("finance")
};
```

Contoh 5: Izinkan alamat IP tertentu

Contoh berikut menunjukkan kebijakan yang hanya mengizinkan alamat IP tertentu.

```
permit(principal, action, resource)
when {
    context.http_request.client_ip == "192.0.2.1"
};
```

Contoh 5a: Blokir alamat IP tertentu

Contoh berikut menunjukkan kebijakan yang akan memblokir alamat IP tertentu.

```
forbid(principal,action,resource)
when {
ip(context.http_request.client_ip).isInRange(ip("192.0.2.1/32"))
};
```

Asisten kebijakan Akses Terverifikasi

Asisten kebijakan Akses Terverifikasi adalah alat di konsol Akses Terverifikasi yang dapat Anda gunakan untuk menguji dan mengembangkan kebijakan Anda. Ini menyajikan kebijakan titik akhir, kebijakan grup, dan konteks kepercayaan di satu layar, tempat Anda dapat menguji dan mengedit kebijakan.

Format konteks kepercayaan bervariasi di berbagai penyedia kepercayaan, dan terkadang administrator Akses Terverifikasi mungkin tidak mengetahui format persis yang digunakan penyedia kepercayaan tertentu. Itulah mengapa sangat membantu untuk melihat konteks kepercayaan, dan kebijakan kelompok dan titik akhir di satu tempat untuk tujuan pengujian dan pengembangan.

Bagian berikut menjelaskan dasar-dasar penggunaan editor kebijakan.

Tugas

- Langkah 1: Tentukan sumber daya Anda
- Langkah 2: Uji dan edit kebijakan
- Langkah 3: Tinjau dan terapkan perubahan

Langkah 1: Tentukan sumber daya Anda

Pada halaman pertama asisten kebijakan, Anda menentukan titik akhir Akses Terverifikasi yang ingin Anda gunakan. Anda juga akan menentukan pengguna (diidentifikasi oleh alamat email), dan secara opsional, nama pengguna dan/atau pengenal perangkat. Secara default, keputusan otorisasi terbaru

diekstraksi dari log Akses Terverifikasi untuk pengguna tertentu. Anda dapat secara opsional memilih mengizinkan atau menolak keputusan terbaru secara khusus.

Terakhir, konteks kepercayaan, keputusan otorisasi, kebijakan titik akhir, dan kebijakan grup semuanya ditampilkan di layar berikutnya.

Untuk membuka asisten kebijakan dan menentukan sumber daya Anda

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instance Akses Terverifikasi, lalu klik ID instans Akses Terverifikasi untuk instance yang ingin Anda gunakan.
3. Pilih Peluncuran asisten kebijakan.
4. Untuk alamat email Pengguna, masukkan alamat email pengguna.
5. Untuk titik akhir Akses Terverifikasi, pilih titik akhir yang ingin Anda edit dan uji kebijakan.
6. (Opsional) Untuk Nama, berikan nama pengguna.
7. (Opsional) Di bawah Pengenal perangkat, berikan pengenal perangkat unik.
8. (Opsional) Untuk hasil Otorisasi, pilih jenis hasil otorisasi terbaru yang ingin Anda gunakan. Secara default, hasil otorisasi terbaru akan digunakan.
9. Pilih Berikutnya.

Langkah 2: Uji dan edit kebijakan

Pada halaman ini Anda akan disajikan dengan informasi berikut untuk bekerja dengan:

- Konteks kepercayaan yang dikirim oleh penyedia kepercayaan Anda untuk pengguna dan (opsional) perangkat yang Anda tentukan pada langkah sebelumnya.
- Kebijakan Cedar untuk titik akhir Akses Terverifikasi yang ditentukan pada langkah sebelumnya.
- Kebijakan Cedar untuk grup Akses Terverifikasi yang menjadi milik titik akhir.

Kebijakan Cedar untuk titik akhir dan grup Akses Terverifikasi dapat diedit di halaman ini, tetapi konteks kepercayaannya statis. Anda sekarang dapat menggunakan halaman ini untuk melihat konteks kepercayaan di samping kebijakan Cedar.

Uji kebijakan terhadap konteks kepercayaan dengan memilih tombol Uji kebijakan, dan hasil otorisasi akan ditampilkan di layar. Anda dapat mengedit kebijakan dan menguji ulang perubahan Anda, mengulangi proses sesuai kebutuhan.

Setelah Anda puas dengan perubahan yang dibuat pada kebijakan, pilih Berikutnya untuk melanjutkan ke layar asisten kebijakan berikutnya.

Langkah 3: Tinjau dan terapkan perubahan

Pada halaman terakhir asisten kebijakan, Anda akan melihat perubahan yang Anda buat pada kebijakan yang disorot agar mudah ditinjau. Anda sekarang dapat meninjaunya untuk terakhir kalinya dan memilih Terapkan perubahan untuk melakukan perubahan.

Anda juga memiliki opsi untuk kembali ke halaman sebelumnya dengan memilih Sebelumnya, atau membatalkan asisten kebijakan sepenuhnya dengan memilih Batal.

Keamanan dalam Akses AWS Terverifikasi

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. Model tanggung jawab bersama menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan dari cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS Cloud. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi secara berkala efektivitas keamanan kami sebagai bagian dari Program Kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku untuk Akses AWS Terverifikasi, lihat AWSLayanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Akses Terverifikasi. Topik berikut menunjukkan cara mengonfigurasi Akses Terverifikasi untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Akses Terverifikasi Anda.

Konten

Perlindungan data dalam Akses AWS Terverifikasi

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Akses AWS Terverifikasi. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk melindungi

infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya lindungi kredensial Akun AWS dan siapkan untuk masing-masing pengguna AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya AWS. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pengelogan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama semua kontrol keamanan bawaan dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Akses Terverifikasi atau lainnya Layanan AWS menggunakan konsol, APIAWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi dalam bergerak

Verified Access mengenkripsi semua data dalam perjalanan dari pengguna akhir ke titik akhir Akses Terverifikasi melalui Internet menggunakan Transport Layer Security (TLS) 1.2 atau yang lebih baru.

Privasi lalu lintas antar jaringan

Anda dapat mengonfigurasi Akses Terverifikasi untuk membatasi akses ke sumber daya tertentu di VPC Anda. Untuk otentikasi berbasis pengguna, Anda juga dapat membatasi akses ke bagian jaringan Anda, berdasarkan grup pengguna yang mengakses titik akhir. Untuk informasi selengkapnya, lihat Kebijakan Akses Terverifikasi.

Enkripsi data saat istirahat untuk Akses AWS Terverifikasi

AWSAkses Terverifikasi mengenkripsi data saat istirahat secara default, menggunakan kunci KMS yang AWS dimiliki. Ketika enkripsi data saat istirahat terjadi secara default, ini membantu mengurangi overhead operasional dan kompleksitas yang terlibat dalam melindungi data sensitif. Pada saat yang sama, ini memungkinkan Anda untuk membangun aplikasi aman yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan. Bagian berikut memberikan rincian tentang bagaimana Akses Terverifikasi menggunakan kunci KMS untuk enkripsi data saat istirahat.

Daftar Isi

Akses Terverifikasi dan kunci KMS

AWSkunci yang dimiliki

Akses Terverifikasi menggunakan kunci KMS untuk mengenkripsi informasi identitas pribadi (PII) secara otomatis. Ini terjadi secara default, dan Anda sendiri tidak dapat melihat, mengelola, menggunakan, atau mengaudit penggunaan kunci yang dimiliki AWS. Namun, Anda tidak perlu mengambil tindakan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda. Untuk informasi selengkapnya, lihat kunci yang AWS dimiliki di Panduan AWS Key Management Service Pengembang.

Meskipun Anda tidak dapat menonaktifkan lapisan enkripsi ini atau memilih jenis enkripsi alternatif, Anda dapat menambahkan lapisan enkripsi kedua di atas kunci enkripsi yang ada AWS dengan memilih kunci yang dikelola pelanggan saat Anda membuat sumber daya Akses Terverifikasi.

Kunci yang dikelola pelanggan

Akses Terverifikasi mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat dan kelola, untuk menambahkan lapisan enkripsi kedua di atas enkripsi default yang ada. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:

- Menetapkan dan memelihara kebijakan utama
- Menetapkan dan memelihara kebijakan dan hibah IAM
- Mengaktifkan dan menonaktifkan kebijakan utama
- Memutar bahan kriptografi kunci
- Menambahkan tanda
- Membuat alias kunci
- Kunci penjadwalan untuk penghapusan

Untuk informasi selengkapnya, lihat Kunci terkelola pelanggan di Panduan AWS Key Management Service Pengembang.

> **Note**
>
> Akses Terverifikasi secara otomatis mengaktifkan enkripsi saat istirahat menggunakan kunci yang AWS dimiliki untuk melindungi data yang dapat diidentifikasi secara pribadi tanpa biaya. Namun, AWS KMS biaya akan berlaku ketika Anda menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang harga, lihat AWS Key Management Serviceharga.

Informasi pengenal pribadi

Tabel berikut merangkum informasi yang dapat diidentifikasi secara pribadi (PII) yang digunakan Akses Terverifikasi, dan bagaimana informasi tersebut dienkripsi.

Tipe data	AWSenkripsi kunci yang dimiliki	Enkripsi kunci yang dikelola pelanggan (Opsional)
`Trust provider (user-type)` Penyedia kepercayaan tipe pengguna berisi opsi OIDC seperti AuthorizationEndpoint,,, UserInfoEndpoint ClientId, dan sebagainya ClientSecret, yang dianggap PII.	Aktif	Aktif
`Trust provider (device-type)` Penyedia kepercayaan tipe perangkat berisi a TenantId, yang dianggap PII.	Aktif	Aktif
`Group policy` Disediakan selama pembuatan atau modifikasi grup Akses Terverifikasi. Berisi aturan untuk mengotorisasi permintaan akses. Mungkin berisi PII seperti nama pengguna dan alamat email, dan sebagainya.	Aktif	Aktif
`Endpoint policy` Disediakan selama pembuatan atau modifikasi titik akhir Akses Terverifikasi. Berisi aturan untuk	Aktif	Aktif

Tipe data	AWSenkripsi kunci yang dimiliki	Enkripsi kunci yang dikelola pelanggan (Opsional)
mengotorisasi permintaan akses. Mungkin berisi PII seperti nama pengguna dan alamat email, dan sebagainya.		

Bagaimana Akses AWS Terverifikasi menggunakan hibah di AWS KMS

Akses Terverifikasi memerlukan hibah untuk menggunakan kunci terkelola pelanggan Anda.

Saat Anda membuat sumber daya Akses Terverifikasi yang dienkripsi dengan kunci terkelola pelanggan, Akses Terverifikasi akan membuat hibah atas nama Anda dengan mengirimkan CreateGrantpermintaan ke. AWS KMS Hibah AWS KMS digunakan untuk memberikan Akses Terverifikasi akses ke kunci yang dikelola pelanggan di akun Anda.

Akses Terverifikasi memerlukan hibah untuk menggunakan kunci terkelola pelanggan Anda untuk operasi internal berikut:

- Kirim permintaan Dekripsi ke AWS KMS untuk mendekripsi kunci data terenkripsi sehingga mereka dapat digunakan untuk mendekripsi data Anda.
- Kirim RetireGrantpermintaan AWS KMS untuk menghapus hibah.

Anda dapat mencabut akses ke hibah, atau menghapus akses layanan ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, Akses Terverifikasi tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, yang memengaruhi operasi yang bergantung pada data tersebut.

Menggunakan kunci terkelola pelanggan dengan Akses Terverifikasi

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakanAWS Management Console, atau AWS KMS API. Ikuti langkah-langkah untuk Membuat kunci terkelola pelanggan simetris di Panduan AWS Key Management Service Pengembang.

Kebijakan utama

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang

menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat Mengelola akses ke kunci yang dikelola pelanggan di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci terkelola pelanggan dengan sumber daya Akses Terverifikasi, operasi API berikut harus diizinkan dalam kebijakan kunci:

- `kms:CreateGrant`— Menambahkan hibah ke kunci yang dikelola pelanggan. Memberikan akses kontrol ke kunci KMS tertentu, yang memungkinkan akses untuk memberikan operasi yang diperlukan Akses Terverifikasi. Untuk informasi selengkapnya tentang Menggunakan Hibah, lihat Panduan AWS Key Management Service Pengembang.

  Hal ini memungkinkan Akses Terverifikasi untuk melakukan hal berikut:

  - Panggilan `GenerateDataKeyWithoutPlainText` untuk menghasilkan kunci data terenkripsi dan menyimpannya, karena kunci data tidak segera digunakan untuk mengenkripsi.
  - Panggilan `Decrypt` untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.
  - Siapkan kepala sekolah yang pensiun untuk memungkinkan layanan. `RetireGrant`
- `kms:DescribeKey`— Memberikan detail kunci yang dikelola pelanggan untuk memungkinkan Akses Terverifikasi memvalidasi kunci.
- `kms:GenerateDataKey`— Memungkinkan Akses Terverifikasi untuk menggunakan kunci untuk mengenkripsi data.
- `kms:Decrypt`— Izinkan Akses Terverifikasi untuk mendekripsi kunci data terenkripsi.

Berikut ini adalah contoh kebijakan kunci yang dapat Anda gunakan untuk Akses Terverifikasi.

```
"Statement" : [
    {
      "Sid" : "Allow access to principals authorized to use Verified Access",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "*"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:GenerateDataKey",
```

```
        "kms:Decrypt"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "kms:ViaService" : "verified-access.region.amazonaws.com",
          "kms:CallerAccount" : "111122223333"
        }
    },
    {
      "Sid": "Allow access for key administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action" : [
        "kms:*"
       ],
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    {
      "Sid" : "Allow read-only access to key metadata to the account",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
      ],
      "Resource" : "*"
    }
  ]
```

Untuk informasi selengkapnya tentang menentukan izin dalam kebijakan, lihat Panduan AWS Key Management ServicePengembang.

Untuk informasi selengkapnya tentang akses kunci pemecahan masalah, lihat Panduan AWS Key Management ServicePengembang.

Menentukan kunci terkelola pelanggan untuk sumber daya Akses Terverifikasi

Anda dapat menentukan kunci yang dikelola pelanggan untuk menyediakan enkripsi lapisan kedua untuk sumber daya berikut:

- Grup Akses Terverifikasi
- Titik akhir Akses Terverifikasi
- Penyedia kepercayaan Akses Terverifikasi

Bila Anda membuat salah satu sumber daya ini menggunakanAWS Management Console, Anda dapat menentukan kunci yang dikelola pelanggan di bagian Enkripsi tambahan -- opsional. Selama proses, pilih kotak centang Sesuaikan pengaturan enkripsi (lanjutan), lalu masukkan ID AWS KMS kunci yang ingin Anda gunakan. Ini juga dapat dilakukan ketika memodifikasi sumber daya yang ada, atau dengan menggunakan file. AWS CLI

> **Note**
>
> Jika kunci terkelola pelanggan yang digunakan untuk menambahkan enkripsi tambahan ke salah satu sumber daya di atas hilang, nilai konfigurasi untuk sumber daya tidak lagi dapat diakses. Namun sumber daya dapat dimodifikasi, dengan menggunakan AWS Management Console orAWS CLI, untuk menerapkan kunci yang dikelola pelanggan baru dan mengatur ulang nilai konfigurasi.

AWSKonteks enkripsi Akses Terverifikasi

Konteks enkripsi adalah kumpulan opsional pasangan kunci-nilai yang berisi informasi kontekstual tambahan tentang data. AWS KMSmenggunakan konteks enkripsi sebagai data otentikasi tambahan untuk mendukung enkripsi yang diautentikasi. Bila Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda menyertakan konteks enkripsi yang sama dalam permintaan.

AWSKonteks enkripsi Akses Terverifikasi

Akses Terverifikasi menggunakan konteks enkripsi yang sama di semua operasi AWS KMS kriptografi, di mana kuncinya `aws:verified-access:arn` dan nilainya adalah sumber daya Amazon Resource Name (ARN). Di bawah ini adalah konteks enkripsi untuk sumber daya Akses Terverifikasi.

Penyedia kepercayaan Akses Terverifikasi

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessTrustProviderId"
}
```

Grup Akses Terverifikasi

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessGroupId"
}
```

Titik akhir Akses Terverifikasi

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessEndpointId"
}
```

Untuk informasi selengkapnya tentang penggunaan konteks enkripsi untuk hibah atau kebijakan, lihat konteks enkripsi di Panduan AWS Key Management Service Pengembang.

Memantau kunci enkripsi Anda untuk Akses AWS Terverifikasi

Saat Anda menggunakan kunci KMS yang dikelola pelanggan dengan sumber daya Akses AWS Terverifikasi, Anda dapat menggunakannya AWS CloudTrailuntuk melacak permintaan yang dikirimkan Akses Terverifikasi. AWS KMS

Contoh berikut adalah AWS CloudTrail peristiwa untuk`CreateGrant`,,,`RetireGrant`, dan `Decrypt` `DescribeKey``GenerateDataKey`, yang memantau operasi KMS yang dipanggil oleh Akses Terverifikasi untuk mengakses data yang dienkripsi oleh kunci KMS yang dikelola pelanggan Anda:

CreateGrant

Saat Anda menggunakan kunci yang dikelola pelanggan untuk mengenkripsi sumber daya Anda, Akses Terverifikasi mengirimkan `CreateGrant` permintaan atas nama Anda untuk mengakses kunci di AWS akun Anda. Hibah yang dibuat oleh Akses Terverifikasi khusus untuk sumber daya yang terkait dengan kunci yang dikelola pelanggan.

Contoh peristiwa berikut mencatat `CreateGrant` operasi:


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T16:27:12Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T16:41:42Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "operations": [
            "Decrypt",
            "RetireGrant",
            "GenerateDataKey"
        ],
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae",
        "constraints": {
            "encryptionContextSubset": {
                "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-0e54f581e2e5c97a2"
```

```
            }
        },
        "granteePrincipal": "verified-access.ca-central-1.amazonaws.com",
        "retiringPrincipal": "verified-access.ca-central-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "e5a050fff9893ba1c43f83fddf61e5f9988f579beaadd6d4ad6d1df07df6048f",
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
    },
    "requestID": "0faa837e-5c69-4189-9736-3957278e6444",
    "eventID": "1b6dd8b8-cbee-4a83-9b9d-d95fa5f6fd08",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RetireGrant

Akses Terverifikasi menggunakan `RetireGrant` operasi untuk menghapus hibah saat Anda menghapus sumber daya.

Contoh peristiwa berikut mencatat `RetireGrant` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
```

```
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T16:42:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T16:47:53Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RetireGrant",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": null,
    "responseElements": {
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
    },
    "additionalEventData": {
        "grantId":
 "b35e66f9bacb266cec214fcaa353c9cf750785e28773e61ba6f434d8c5c7632f"
    },
    "requestID": "7d4a31c2-d426-434b-8f86-336532a70462",
    "eventID": "17edc343-f25b-43d4-bbff-150d8fff4cf8",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management"
}
```

Decrypt

Akses Terverifikasi memanggil `Decrypt` operasi untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.

Contoh peristiwa berikut mencatat `Decrypt` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:47:05Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e",
```

```
        "encryptionContext": {
            "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-00f20a4e455e9340f",
            "aws-crypto-public-key": "AkK+vi1W/
acBKv7OR8p2DeUrA8EgpTffSrjBqNucODuBYhyZ3hlMuYYJz9x7CwQWZw=="
        }
    },
    "responseElements": null,
    "requestID": "2e920fd3-f2f6-41b2-a5e7-2c2cb6f853a9",
    "eventID": "3329e0a3-bcfb-44cf-9813-8106d6eee31d",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DescribeKey

Akses Terverifikasi menggunakan `DescribeKey` operasi untuk memverifikasi apakah kunci terkelola pelanggan yang terkait dengan sumber daya Anda ada di akun dan Wilayah.

Contoh peristiwa berikut mencatat `DescribeKey` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
```

```
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:46:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
    },
    "responseElements": null,
    "requestID": "5b127082-6691-48fa-bfb0-4d40e1503636",
    "eventID": "ffcfc2bb-f94b-4c00-b6fb-feac77daff2a",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


GenerateDataKey

Contoh peristiwa berikut mencatat `GenerateDataKey` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:46:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-00f20a4e455e9340f",
            "aws-crypto-public-key": "A/ATGxaYatPUlOtM+l/mfDndkzHUmX5Hav+29IlIm
+JRBKFuXf24ulztmOIsqFQliw=="
        },
        "numberOfBytes": 32,
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
    },
    "responseElements": null,
    "requestID": "06535808-7cce-4ae1-ab40-e3afbf158a43",
```

```
    "eventID": "1ce79601-5a5e-412c-90b3-978925036526",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Manajemen identitas dan akses untuk Akses AWS Terverifikasi

(IAM) AWS Identity and Access Management adalah Layanan AWS yang membantu seorang administrator dalam mengendalikan akses ke sumber daya AWS secara aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Akses Terverifikasi. IAM adalah sebuah layanan Layanan AWS yang dapat Anda gunakan tanpa dikenakan biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola kebijakan menggunakan akses
- Cara Kerja Akses AWS Terverifikasi dengan IAM
- Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS
- Memecahkan masalah Identitas dan akses Akses AWS Terverifikasi
- Menggunakan peran terkait layanan untuk Akses Terverifikasi
- AWSkebijakan terkelola untuk Akses AWS Terverifikasi

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Akses Terverifikasi.

Pengguna layanan — Jika Anda menggunakan layanan Akses Terverifikasi untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensyal dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Akses Terverifikasi untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Akses Terverifikasi, lihatMemecahkan masalah Identitas dan akses Akses AWS Terverifikasi.

Administrator layanan — Jika Anda bertanggung jawab atas sumber daya Akses Terverifikasi di perusahaan Anda, Anda mungkin memiliki akses penuh ke Akses Terverifikasi. Tugas Anda adalah menentukan fitur dan sumber daya Akses Terverifikasi mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Akses Terverifikasi, lihatCara Kerja Akses AWS Terverifikasi dengan IAM.

Administrator IAM — Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Akses Terverifikasi. Untuk melihat contoh kebijakan berbasis identitas Akses Terverifikasi yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS

Mengautentikasi dengan identitas

Autentikasi merupakan cara Anda untuk masuk ke AWS dengan menggunakan kredensial identitas Anda. Anda harus terautentikasi (masuk keAWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk ke AWS sebagai identitas terfederasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Para pengguna (Pusat Identitas IAM), otentikasi sign-on tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda merupakan contoh identitas terfederasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas dengan menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil suatu peran.

Tergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal akses AWS. Untuk informasi selengkapnya tentang masuk ke AWS, silakan lihat Cara masuk ke Akun AWS Anda di Panduan Pengguna AWS Sign-In.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan peralatan AWS, maka Anda harus menandatangani sendiri permintaan tersebut. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, silakan lihat Menandatangani permintaan API AWS di Panduan Pengguna IAM.

Terlepas dari metode autentikasi yang Anda gunakan, Anda mungkin juga diminta untuk menyediakan informasi keamanan tambahan. Sebagai contoh, AWS menyarankan supaya Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, silakan lihat Autentikasi multi-faktor di Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS di Panduan Pengguna IAM.

Pengguna root Akun AWS

Ketika Anda membuat Akun AWS, Anda memulai dengan satu identitas masuk yang memiliki akses ke semua Layanan AWS dan sumber daya di akun tersebut. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk ke alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, silakan lihat Tugas yang memerlukan kredensial pengguna root di Panduan Pengguna IAM.

Identitas terfederasi

Praktik terbaiknya berupa, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial temporer.

Identitas terfederasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, dikenal sebagai AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran memberikan kredensial temporer.

Untuk pengelolaan akses terpusat, kami sarankan Anda menggunakanAWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS Anda dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, silakan lihat Apakah Pusat Identitas IAM itu? di User Guide AWS IAM Identity Center.

Pengguna dan Grup IAM

Pengguna IAM adalah identitas dalam Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Apabila memungkinkan, kami menyarankan untuk mengandalkan pada kredensial temporer alih-alih membuat pengguna IAM yang memiliki kredensyal jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami menyarankan Anda memutar kunci akses. Untuk informasi selengkapnya, silakan lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang di Panduan Pengguna IAM.

Grup IAM adalah identitas yang menerangkan secara spesifik kumpulan pengguna IAM. Anda tidak dapat masuk sebagai kelompok. Anda dapat menggunakan grup untuk menerangkan secara spesifik izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Sebagai contoh, Anda dapat memiliki grup yang diberi nama AdminIAM dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial temporer. Untuk mempelajari selengkapnya, silakan lihat Kapan harus membuat pengguna IAM (alih-alih peran) di Panduan Pengguna IAM.

Peran IAM

Peran IAM merupakan identitas dalam Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat menggunakan peran IAM untuk sementara dalam AWS Management Console dengan berganti peran. Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, silakan lihat menggunakan peran IAM di Panduan Pengguna IAM.

IAM role dengan kredensial temporer berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas terfederasi, Anda harus membuat sebuah peran dan menentukan izin untuk peran tersebut. Ketika identitas gabungan terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran-peran untuk federasi, silakan lihat Membuat sebuah peran untuk Penyedia Identitas pihak ketiga di Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi serangkain izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengkorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, silakan lihat Rangkaian izin di Panduan Pengguna AWS IAM Identity Center.
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM untuk sementara mengambil izin berbeda untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) di akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, pada beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan suatu peran sebagai proksi). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat Bagaimana peran IAM role berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan – Sebagian Layanan AWS menggunakan fitur di Layanan AWS lainnya. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, lazim pada layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran tertaut layanan.
  - Sesi akses maju (FAS) – Ketika Anda menggunakan pengguna IAM atau peran IAM untuk melakukan tindakan-tindakan di AWS, Anda akan dianggap sebagai seorang pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat Meneruskan sesi akses.
  - Peran layanan – Sebuah peran layanan adalah sebuah peran IAM yang dijalankan oleh suatu layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat,

memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, silakan lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

- Peran tertaut layanan – Peran tertaut layanan adalah tipe peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.

- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan peran IAM untuk mengelola kredensial temporer untuk aplikasi yang berjalan di instans EC2 dan mengajukan permintaan AWS CLI atau API AWS. Cara ini lebih baik daripada menyimpan kunci akses dalam instans EC2. Untuk menugaskan sebuah peran AWS ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda dapat membuat sebuah profil instans yang dilampirkan ke instans. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, silakan lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 di Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, silakan lihat Kapan harus membuat peran IAM (alih-alih pengguna) di Panduan Pengguna IAM.

Mengelola kebijakan menggunakan akses

Anda mengendalikan akses di AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya AWS. Kebijakan adalah objek di AWS yang, ketika terkait dengan identitas atau sumber daya, akan menentukan izinnya. AWS mengevaluasi kebijakan-kebijakan tersebut ketika seorang pengguna utama (pengguna, root user, atau sesi peran) mengajukan permintaan. Izin dalam kebijakan menentukan apakah permintaan diberikan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, silakan lihat Gambaran Umum kebijakan JSON di Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Secara bawaan, para pengguna dan peran tidak memiliki izin. Untuk mengabulkan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat

membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk pengoperasiannya. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole` . Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau APIAWS.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline ditanam secara langsung ke pengguna tunggal, grup, atau peran. Kebijakan terkelola adalah kebijakan yang berdiri sendiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS Anda. Kebijakan terkelola mencakup kebijakan terkelola AWS dan kebijakan terkelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, silakan lihat Memilih antara kebijakan terkelola dan kebijakan inline diPanduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentkan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan terkelola AWS dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh-contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Ringkas Amazon.

Tipe-tipe kebijakan lain

AWS mendukung tipe kebijakan tambahan, yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batas izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini menindihi izin. Untuk informasi selengkapnya tentang batasan izin, silakan lihat Batasan izin untuk entitas IAM di Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan secara terpusat mengelola beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau ke semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organisasi dan SCP, silakan lihat Cara kerja SCP di Panduan Pengguna AWS Organizations.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini menindihi izin. Untuk informasi selengkapnya, silakan lihat Kebijakan sesi di Panduan Pengguna IAM.

Berbagai tipe kebijakan

Ketika beberapa tipe kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan ketika beberapa tipe kebijakan dilibatkan, silakan lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Cara Kerja Akses AWS Terverifikasi dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Akses Terverifikasi, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Akses Terverifikasi.

Fitur IAM yang dapat Anda gunakan dengan Akses AWS Terverifikasi

Fitur IAM	Dukungan Akses Terverifikasi
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACL	Tidak
ABAC (tag dalam kebijakan)	Parsial
Kredensial temporer	Ya
Izin-izin pengguna utama	Ya
Peran layanan	Tidak
Peran tertaut layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Akses Terverifikasi dan AWS layanan lainnya dengan sebagian besar fitur IAM, lihat AWSlayanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Akses Terverifikasi

Mendukung kebijakan berbasis identitas	Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta persyaratan yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik pengguna utama dalam sebuah kebijakan berbasis identitas karena pengguna utama berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, silakan lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Akses Terverifikasi

Untuk melihat contoh kebijakan berbasis identitas Akses Terverifikasi, lihat. Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS

Kebijakan berbasis sumber daya dalam Akses Terverifikasi

Mendukung kebijakan berbasis sumber daya	Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentkan tindakan apa yang dapat

dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika pengguna utama dan sumber daya berada dalam Akun AWS yang berbeda, Administrator IAM di akun tepercaya juga harus memberikan izin kepada entitas pengguna utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, silakan lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Tindakan kebijakan untuk Akses Terverifikasi

Mendukung tindakan kebijakan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan-tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan-tindakan kebijakan biasanya memiliki nama yang sama sebagaimana operasi API AWS yang dikaitkan padanya. Ada beberapa pengecualian, misalnya tindakan yang memiliki izin saja yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam sebuah kebijakan. Tindakan-tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Untuk melihat daftar tindakan Akses Terverifikasi, lihat Tindakan yang Ditentukan oleh Amazon EC2 di Referensi Otorisasi Layanan.

Tindakan kebijakan di Akses Terverifikasi menggunakan awalan berikut sebelum tindakan:

```
ec2
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut dengan koma.

```
"Action": [
    "ec2:action1",
    "ec2:action2"
]
```

Untuk melihat contoh kebijakan berbasis identitas Akses Terverifikasi, lihat. Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS

Sumber daya kebijakan untuk Akses Terverifikasi

Mendukung sumber daya kebijakan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan entah elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan-tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk mengindikasikan bahwa pernyataan tersebut berlaku bagi semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya Akses Terverifikasi dan ARNnya, lihat Sumber Daya yang Ditentukan oleh Amazon EC2 di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana

yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang Ditentukan oleh Amazon EC2.

Untuk melihat contoh kebijakan berbasis identitas Akses Terverifikasi, lihat. Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS

Kunci kondisi kebijakan untuk Akses Terverifikasi

Mendukung kunci-kunci persyaratan kebijakan spesifik layanan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan syarat yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator syarat, misalnya sama dengan atau kurang dari, untuk mencocokkan syarat dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya dengan menggunakan operasi `AND` yang logis. Jika Anda menentukan beberapa niilai untuk satu kunci persyaratan, maka AWS akan mengevaluasi syarat tersebut menggunakan operasi `OR` yang logis. Semua persyaratan harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan syarat. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat Elemen kebijakan IAM: variabel dan tag di Panduan Pengguna IAM.

AWS mendukung kunci-kunci syarat global dan kunci-kunci syarat spesifik layanan. Untuk melihat semua kunci persyaratan global AWS, silakan lihat kunci konteks syarat global AWS di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Akses Terverifikasi, lihat Kunci Kondisi untuk Amazon EC2 di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya mana untuk gunakan kunci syarat, lihat Tindakan yang Ditentukan oleh Amazon EC2.

Untuk melihat contoh kebijakan berbasis identitas Akses Terverifikasi, lihat. Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS

ACL dalam Akses Terverifikasi

Mendukung ACL	Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Akses Terverifikasi

Mendukung ABAC (tag dalam kebijakan)	Parsial

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Di AWS, atribut-atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak sumber daya AWS. Pemberian tag ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi-operasi ketika tag milik pengguna utama cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi dimana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen syarat dari sebuh kebijakan dengan menggunakan kunci-kunci persyaratan `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci-kunci persyaratan untuk setiap jenis sumber daya, maka nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci persyaratan untuk hanya beberapa jenis sumber daya, maka nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, silakan lihat Apa itu ABAC? di Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, silakan lihat Menggunakan kontrol akses berbasis atribut (ABAC) di Panduan Pengguna IAM.

Menggunakan kredensyal sementara dengan Akses Terverifikasi

Mendukung kredensial temporer	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk dengan menggunakan kredensial temporer. Sebagai informasi tambahan, termasuk tentang Layanan AWS mana saja yang berfungsi dengan kredensial temporer, silakan lihat Layanan AWS yang berfungsi dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensial temporer jika Anda masuk ke AWS Management Console dengan menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Sebagai contoh, ketika Anda mengakses AWS dengan menggunakan tautan masuk tunggal (SSO) milik perusahaan Anda, proses itu secara otomatis akan membuat kredensial temporer. Anda juga akan secara otomatis membuat kredensial temporer ketika Anda masuk ke konsol sebagai seorang pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang peralihan peran, silakan lihat Peralihan peran (konsol) di Panduan Pengguna IAM.

Anda dapat secara manual membuat kredensial temporer menggunakan AWS CLI atau API AWS. Anda kemudian dapat menggunakan kredensial temporer tersebut untuk mengakses AWS. AWS menyarankan agar Anda secara dinamis membuat kredensial temporer alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, silakan lihat Kredensial keamanan temporer di IAM.

Izin utama lintas layanan untuk Akses Terverifikasi

Mendukung sesi akses maju (FAS)	Ya

Saat Anda menggunakan pengguna IAM atau peran IAM untuk mengerjakan tindakan di AWS, Anda akan dianggap sebagai pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat Meneruskan sesi akses.

Peran layanan untuk Akses Terverifikasi

Mendukung peran layanan	Tidak

Peran layanan adalah sebuah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

Peran terkait layanan untuk Akses Terverifikasi

Mendukung peran yang terhubung dengan layanan	Ya

Peran yang tertaut layanan adalah jenis peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan Akses Terverifikasi, lihat. Menggunakan peran terkait layanan untuk Akses Terverifikasi

Contoh kebijakan berbasis identitas untuk Akses Terverifikasi AWS

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Akses Terverifikasi. Pengguan dan peran tersebut juga tidak dapat melakukan tugas dengan menggunakan API AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS. Untuk mengabulkan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, silakan lihat Membuat kebijakan IAM di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Akses Terverifikasi, termasuk format ARN untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk Amazon EC2 di Referensi Otorisasi Layanan.

Topik

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Akses Terverifikasi di akun Anda. Tindakan ini mengenakan biaya kepada Anda Akun AWS. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan terkelola AWS dan beralih ke izin dengan hak akses paling rendah – Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan terkelola AWS yang memberikan izin untuk banyak kasus penggunaan umum. Kebijakan tedapat di Akun AWS Anda. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan AWS yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, silakan lihat kebijakan-kebijakan terkelola AWS atau kebijakan-kebijakan terkelola AWS untuk fungsi tugas di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan pengguna IAM untuk mengajukan izin, silakan lihat Kebijakan dan izin di IAM di Panduan Pengguna IAM.
- Gunakan syarat dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu syarat ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan syarat untuk memberi akses ke tindakan layanan jika digunakan melalui Layanan AWS yang spesifk, sepertiAWS CloudFormation. Untuk informasi selengkapnya, silakan lihat Elemen kebijakan JSON IAM: Syarat di Panduan Pengguna IAM.
- Gunakan Analizer Akses IAM untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – Analizer Akses IAM memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM.

Analizer Akses IAM menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, silakan lihat validasi kebijakan Analizer Akses IAM di Panduan Pengguna IAM.

- Memerlukan autentikasi multi-faktor (MFA) – Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Akun AWS Anda, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan syarat MFA pada kebijakan Anda. Untuk informasi selengkapnya, silakan lihat Mengonfigurasi akses API yang diproteksi MFA di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, silakan lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Kebijakan untuk membuat instance Akses Terverifikasi

Untuk membuat instance Akses Terverifikasi, prinsipal IAM perlu menambahkan pernyataan tambahan ini ke kebijakan IAM mereka.

```
{
    "Effect": "Allow",
    "Action": "verified-access:AllowVerifiedAccess",
    "Resource": "*"
}
```

> **Note**
>
> `verified-access:AllowVerifiedAccess`adalah API virtual khusus aksi.
> Itu tidak mendukung otorisasi berbasis kunci sumber daya, tag, atau kondisi.
> Gunakan otorisasi berbasis kunci sumber daya, tag, atau kondisi pada tindakan API.
> `ec2:CreateVerifiedAccessInstance`

Contoh kebijakan untuk membuat instance Akses Terverifikasi. Dalam contoh ini, `123456789012` adalah nomor AWS rekening dan `us-east-1` merupakan AWS wilayah.

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

```
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVerifiedAccessInstance",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/*"
        },
        {
            "Effect": "Allow",
            "Action": "verified-access:AllowVerifiedAccess",
            "Resource": "*"
        }
    ]
}
```

Perbolehkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda dapat membuat kebijakan yang mengizinkan para pengguna IAM untuk melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan pada konsol atau secara terprogram menggunakan API AWS CLI atau AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
```

```
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Memecahkan masalah Identitas dan akses Akses AWS Terverifikasi

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Akses Terverifikasi dan IAM.

Masalah

- Saya tidak berwenang untuk melakukan tindakan di Akses Terverifikasi
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Akses Terverifikasi saya

Saya tidak berwenang untuk melakukan tindakan di Akses Terverifikasi

Jika Anda menerima pesan galat bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh galat berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `ec2:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `ec2:`*`GetWidget`*.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Akses Terverifikasi.

Sebagian Layanan AWS mengizinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut-layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Akses Terverifikasi. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Akses Terverifikasi saya

Anda dapat membuat peran yang dapat digunakan para pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses kepada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mengetahui apakah Akses Terverifikasi mendukung fitur ini, lihatCara Kerja Akses AWS Terverifikasi dengan IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, silakan lihat Menyediakan akses ke pengguna IAM di Akun AWS lainnya yang Anda miliki di Panduan Pengguna IAM.

- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke pihak ketiga Akun AWS, silakan lihat Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, silakan lihat Menyediakan akses ke pengguna terautentikasi eksternal (gabungan identitas) di Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan IAM role dan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Menggunakan peran terkait layanan untuk Akses Terverifikasi

AWSAkses Terverifikasi menggunakan AWS Identity and Access Management peran terkait layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Akses Terverifikasi. Peran terkait layanan ditentukan sebelumnya oleh Akses Terverifikasi dan mencakup semua izin yang diperlukan layanan untuk menelepon orang lain Layanan AWS atas nama Anda.

Peran tertaut layanan membuat pengaturan Akses Terverifikasi menjadi lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Akses Terverifikasi mendefinisikan izin peran terkait layanannya, dan kecuali ditentukan lain, hanya Akses Terverifikasi yang dapat mengambil perannya. Izin yang ditetapkan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin ini tidak dapat dilampirkan ke entitas IAM lainnya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat Layanan AWS yang Bekerja bersama IAM dan mencari layanan yang memiliki Ya dalam Peran Terkait Layanan. Pilih Ya dengan tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Akses Terverifikasi

Akses Terverifikasi menggunakan peran terkait layanan yang diberi nama AWSServiceRoleForVPCVerifiedAccessuntuk menyediakan sumber daya di akun Anda yang diperlukan untuk menggunakan layanan.

AWSServiceRoleForVPCVerifiedAccess peran terkait layanan memercayakan layanan berikut untuk menjalankan peran tersebut:

- `verified-access.amazonaws.com`

Kebijakan izin peran, bernama AWSVPCVerifiedAccessServiceRolePolicy, memungkinkan Akses Terverifikasi untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan `ec2:CreateNetworkInterface` pada semua subnet dan grup keamanan, serta semua antarmuka jaringan dengan tag `VerifiedAccessManaged=true`
- Tindakan `ec2:CreateTags` pada semua antarmuka jaringan pada waktu pembuatan
- Tindakan `ec2:DeleteNetworkInterface` pada semua antarmuka jaringan dengan tag `VerifiedAccessManaged=true`
- Tindakan `ec2:ModifyNetworkInterfaceAttribute` pada semua grup keamanan dan semua antarmuka jaringan dengan tag `VerifiedAccessManaged=true`

Anda juga dapat melihat izin untuk kebijakan ini di AWS Management Console AWSVPCVerifiedAccessServiceRolePolicy, atau Anda dapat melihat AWSVPCVerifiedAccessServiceRolePolicykebijakan di Panduan Referensi Kebijakan AWS Terkelola.

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Akses Terverifikasi

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda memanggil CreateVerifiedAccessEndpointAWS Management Console,, atau AWS APIAWS CLI, Akses Terverifikasi akan membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda menelepon CreateVerifiedAccessEndpointsekali lagi, Akses Terverifikasi akan membuat peran tertaut layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk Akses Terverifikasi

Akses Terverifikasi tidak memungkinkan Anda untuk mengedit peran AWSServiceRoleForVPCVerifiedAccessterkait layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran yang terkait dengan layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Akses Terverifikasi

Anda tidak perlu menghapus peran AWSServiceRoleForVPCVerifiedAccess secara manual. Saat Anda memanggil DeleteVerifiedAccessEndpoint, API AWS Management ConsoleAWS CLI, atau AWS API, Akses Terverifikasi membersihkan sumber daya dan menghapus peran terkait layanan untuk Anda.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, AWS CLI, atau AWS API untuk menghapus peran terkait layanan AWSServiceRoleForVPCVerifiedAccess. Untuk informasi selengkapnya, lihat Menghapus peran tertaut layanan dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk peran terkait layanan Akses Terverifikasi

Akses Terverifikasi mendukung penggunaan peran terkait layanan di semua Wilayah AWS tempat layanan tersedia. Untuk informasi selengkapnya, lihat AWSWilayah dan titik akhir.

AWSkebijakan terkelola untuk Akses AWS Terverifikasi

Kebijakan terkelola AWS adalah kebijakan mandiri yang dibuat dan oleh dilakukan AWS. Kebijakan terkelola AWS dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan terkelola AWS mungkin tidak memberikan izin hak akses paling rendah untuk kasus penggunaan khusus Anda karena tersedia untuk digunakan semua pelanggan AWS. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ada dalam kebijakan-kebijakan terkelola AWS. Jika AWS memperbarui izin yang ditentukan dalam sebuah kebijakan terkelola AWS, maka pembaruan itu akan mempengaruhi semua identitas pengguna utama (pengguna, grup, dan peran) yang terkait dengan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan terkelola AWS saat sebuah Layanan AWS baru diluncurkan atau operasi API baru tersedia untuk layanan yang sudah ada.

Untuk informasi selengkapnya, silakan lihat kebijakan terkelola AWS di Panduan Pengguna IAM.

Kebijakan terkelola AWS: AWSVPCVerifiedAccessServiceRolePolicy

Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan Akses Terverifikasi untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan. Untuk melihat izin kebijakan ini, Anda dapat melihat AWSVPCVerifiedAccessServiceRolePolicydiAWS Management Console, atau Anda dapat melihat AWSVPCVerifiedAccessServiceRolePolicykebijakan di Panduan Referensi Kebijakan AWS Terkelola.

Pembaruan Akses Terverifikasi ke kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Akses Terverifikasi sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen Akses Terverifikasi.

Perubahan	Deskripsi	Tanggal
AWSVPCVerifiedAccessServiceRolePolicy- Kebijakan diperbarui	Akses Terverifikasi memperbarui kebijakan terkelola untuk menyertakan deskripsi semua tindakan di bawah bidang “sid”.	17 November 2023
AWSVPCVerifiedAccessServiceRolePolicy- Kebijakan diperbarui	Akses Terverifikasi memperbarui kebijakan terkelola untuk menambahkan sumber daya grup keamanan ke `ec2:CreateNetworkInterface` izin.	31 Mei 2023
AWSVPCVerifiedAccessServiceRolePolicy- Kebijakan baru	Akses Terverifikasi menambahkan kebijakan baru untuk memungkinkannya menyediakan sumber daya di akun Anda yang diperlukan untuk menggunakan layanan.	29 November 2022

Perubahan	Deskripsi	Tanggal
Akses Terverifikasi mulai melacak perubahan	Akses Terverifikasi mulai melacak perubahan untuk kebijakan yang AWS dikelola.	29 November 2022

Validasi kepatuhan untuk Akses AWS Terverifikasi

Akses Terverifikasi AWSdapat dikonfigurasi untuk mendukung kepatuhan Federal Information Processing Standards (FIPS). Untuk info dan detail selengkapnya tentang pengaturan kepatuhan FIPS untuk Akses Terverifikasi, buka. Kepatuhan FIPS untuk Akses Terverifikasi

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan khusus, lihat Layanan AWS di Scope oleh Program Program Kepatuhan yang Anda minati. Untuk informasi umum, silakan lihat Program Kepatuhan AWS .

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.
- Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

  > **Note**
  >
  > Tidak semua Layanan AWS memenuhi syarat HIPAA. Untuk informasi selengkapnya, lihat Referensi Layanan yang Memenuhi Syarat HIPAA.

- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.

- Panduan Kepatuhan Pelanggan AWS – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- AWS Security Hub – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dadlam AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda dan untuk memeriksa kepatuhan terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- AWS Audit Manager – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan bagaimana Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

Ketahanan dalam AWS Akses Terverifikasi

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Dengan Availability Zone, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami fail over antar zona tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, toleran terhadap kegagalan, dan dapat diukur skalanya jika dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur Global AWS.

Selain infrastruktur AWS global, Verified Access menawarkan fitur berikut untuk membantu mendukung kebutuhan ketersediaan tinggi Anda.

Beberapa subnet untuk ketersediaan tinggi

Saat Anda membuat titik akhir Akses Terverifikasi tipe penyeimbang beban, Anda dapat mengaitkan beberapa subnet ke titik akhir. Setiap subnet yang Anda kaitkan dengan titik akhir harus dimiliki oleh

Availability Zone yang berbeda. Dengan mengaitkan beberapa subnet, Anda dapat memastikan ketersediaan tinggi dengan menggunakan beberapa Availability Zone.

Memantau Akses AWS Terverifikasi

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa Akses AWS Terverifikasi. AWSmenyediakan alat pemantauan berikut untuk mengawasi Akses Terverifikasi, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- Log akses - Tangkap informasi terperinci tentang permintaan untuk mengakses aplikasi. Untuk informasi selengkapnya, lihat the section called “Log Akses Terverifikasi”.
- AWS CloudTrail— Menangkap panggilan API dan kejadian terkait yang dilakukan atas nama Anda Akun AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun yang memanggil AWS, alamat IP asal panggilan dilakukan, dan waktu panggilan terjadi. Untuk informasi selengkapnya, lihat the section called “Log CloudTrail”.

Log Akses Terverifikasi

Setelah AWS Verified Access mengevaluasi setiap permintaan akses, ia mencatat semua upaya akses. Ini memberikan visibilitas terpusat ke dalam akses aplikasi dan membantu Anda dengan cepat menanggapi insiden keamanan dan permintaan audit. Akses Terverifikasi mendukung format pencatatan Open Cybersecurity Schema Framework (OCSF).

Ketika Anda mengaktifkan logging, Anda akan perlu untuk mengkonfigurasi tujuan untuk log yang akan dikirim. Prinsipal IAM yang digunakan untuk mengonfigurasi tujuan logging harus memiliki izin tertentu agar logging berfungsi dengan baik. Izin IAM yang diperlukan untuk setiap tujuan pencatatan dapat dilihat di bagian ini. Izin pencatatan Akses Terverifikasi mendukung tujuan berikut untuk menerbitkan log akses:

- Grup CloudWatch log Amazon Logs
- Bucket Amazon S3
- Aliran pengiriman Amazon Data Firehose

Daftar Isi

- Termasuk konteks kepercayaan
- Contoh entri log untuk log Akses Terverifikasi

Versi logging

Secara default, sistem logging Akses Terverifikasi menggunakan Open Cybersecurity Schema Framework (OCSF) versi 0.1. Contoh log menggunakan versi 0.1 dapat dilihat di Contoh OCSF versi 0.1 bagian.

Versi logging terbaru kompatibel dengan OCSF versi 1.0.0-rc.2. Rincian spesifik tentang skema dapat ditemukan di sini Skema OCSF. Contoh log menggunakan versi 1.0.0-rc.2 dapat dilihat di bagian. Contoh OCSF versi 1.0.0-rc.2

Tingkatkan versi logging

Jika Anda ingin memutakhirkan versi logging yang digunakan, ikuti prosedur di bawah ini.

Untuk meningkatkan versi logging menggunakan konsol

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi yang sesuai.
4. Pada tab konfigurasi pencatatan instans Akses Terverifikasi, pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.
5. Pilih ocsf-1.0.0-rc.2 dari daftar drop-down Perbarui versi log.
6. Pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.

Untuk meningkatkan versi logging menggunakan AWS CLI

Gunakan perintah modify-verified-access-instance-logging-configuration.

Izin pencatatan

Prinsipal IAM yang digunakan untuk mengonfigurasi tujuan logging harus memiliki izin tertentu agar logging berfungsi dengan baik. Di bawah ini Anda dapat melihat izin yang diperlukan untuk setiap tujuan pencatatan.

Untuk pengiriman ke CloudWatch Log:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`pada instance Akses Terverifikasi
- `logs:CreateLogDelivery`,`logs:DeleteLogDelivery`,`logs:GetLogDelivery`,`logs:ListLogDe` dan `logs:UpdateLogDelivery` pada semua sumber daya
- `logs:DescribeLogGroups`,`logs:DescribeResourcePolicies`, dan `logs:PutResourcePolicy` pada grup log tujuan

Untuk pengiriman ke Amazon S3:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`pada instance Akses Terverifikasi
- `logs:CreateLogDelivery`,`logs:DeleteLogDelivery`,`logs:GetLogDelivery`,`logs:ListLogDe` dan `logs:UpdateLogDelivery` pada semua sumber daya
- `s3:GetBucketPolicy`dan `s3:PutBucketPolicy` di ember tujuan

Untuk pengiriman ke Firehose:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`pada instance Akses Terverifikasi
- `firehose:TagDeliveryStream` di semua sumber daya
- `iam:CreateServiceLinkedRole` di semua sumber daya
- `logs:CreateLogDelivery`,`logs:DeleteLogDelivery`,`logs:GetLogDelivery`,`logs:ListLogDe` dan `logs:UpdateLogDelivery` pada semua sumber daya

Mengaktifkan atau menonaktifkan log

Ketika Anda mengaktifkan logging, Anda akan perlu untuk mengkonfigurasi tujuan untuk log yang akan dikirim. Prinsipal IAM yang digunakan untuk mengonfigurasi tujuan logging harus memiliki izin tertentu agar logging berfungsi dengan baik. Izin IAM yang diperlukan untuk setiap tujuan pencatatan dapat dilihat di bagian ini. Izin pencatatan

Daftar Isi

- Nonaktifkan log akses

Aktifkan log akses

Untuk mengaktifkan log Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi.
4. Pada tab konfigurasi pencatatan instans Akses Terverifikasi, pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.
5. (Opsional) Untuk menyertakan data kepercayaan yang dikirim dari penyedia kepercayaan di log, lakukan hal berikut:
   a. Pilih ocsf-1.0.0-rc.2 dari daftar drop-down Perbarui versi log.
   b. Pilih Sertakan konteks kepercayaan.
6. Lakukan salah satu dari cara berikut:
   - Aktifkan Kirim ke CloudWatch Log Amazon. Pilih grup log tujuan.
   - Aktifkan Kirim ke Amazon S3. Masukkan nama, pemilik, dan awalan bucket tujuan.
   - Nyalakan Kirim ke Firehose. Pilih aliran pengiriman tujuan.
7. Pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.

Untuk mengaktifkan log Akses Terverifikasi menggunakan AWS CLI

Gunakan perintah modify-verified-access-instance-logging-configuration.

Nonaktifkan log akses

Anda dapat menonaktifkan log akses untuk instans Akses Terverifikasi kapan saja. Setelah Anda menonaktifkan log akses, data log Anda tetap berada di tujuan log Anda sampai Anda menghapusnya.

Untuk menonaktifkan log Akses Terverifikasi

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.

3. Pilih instance Akses Terverifikasi.
4. Pada tab konfigurasi pencatatan instans Akses Terverifikasi, pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.
5. Matikan pengiriman log.
6. Pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.

Untuk menonaktifkan log Akses Terverifikasi menggunakan AWS CLI

Gunakan perintah modify-verified-access-instance-logging-configuration.

Termasuk konteks kepercayaan

Konteks kepercayaan yang dikirim dari penyedia kepercayaan Anda secara opsional dapat disertakan dalam log Akses Terverifikasi Anda. Ini bisa sangat berguna saat mendefinisikan kebijakan yang mengizinkan atau menolak akses ke aplikasi Anda. Setelah diaktifkan, konteks kepercayaan akan ditemukan di log di bawah `data` bidang. Jika dinonaktifkan, `data` bidang akan diatur ke`null`. Untuk mengonfigurasi Akses Terverifikasi untuk menyertakan konteks kepercayaan dalam log, ikuti prosedur di bawah ini.

> **Note**
>
> Menyertakan konteks kepercayaan dalam log Akses Terverifikasi Anda memerlukan peningkatan ke versi `ocsf-1.0.0-rc.2` logging terbaru. Prosedur di bawah ini mengasumsikan bahwa Anda sudah mengaktifkan logging. Jika itu tidak benar, lihat Aktifkan log akses prosedur lengkapnya.

Daftar Isi

Aktifkan konteks kepercayaan

Untuk menyertakan konteks kepercayaan dalam log Akses Terverifikasi menggunakan konsol

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.

3. Pilih instance Akses Terverifikasi yang sesuai.
4. Pada tab konfigurasi pencatatan instans Akses Terverifikasi, pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.
5. Pilih ocsf-1.0.0-rc.2 dari daftar drop-down Perbarui versi log.
6. Aktifkan Sertakan konteks kepercayaan.
7. Pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.

Untuk menyertakan konteks kepercayaan dalam log Akses Terverifikasi menggunakan AWS CLI

Gunakan perintah modify-verified-access-instance-logging-configuration.

Nonaktifkan konteks kepercayaan

Jika Anda tidak lagi ingin memasukkan konteks kepercayaan dalam log, Anda dapat menghapusnya dengan prosedur di bawah ini.

Untuk menghapus konteks kepercayaan dari log Akses Terverifikasi menggunakan konsol

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di panel navigasi, pilih instans Akses Terverifikasi.
3. Pilih instance Akses Terverifikasi yang sesuai.
4. Pada tab konfigurasi pencatatan instans Akses Terverifikasi, pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.
5. Matikan Sertakan konteks kepercayaan.
6. Pilih Ubah konfigurasi pencatatan instans Akses Terverifikasi.

Untuk menghapus konteks kepercayaan dari log Akses Terverifikasi menggunakan AWS CLI

Gunakan perintah modify-verified-access-instance-logging-configuration.

Contoh entri log untuk log Akses Terverifikasi

Berikut ini adalah contoh entri log.

Daftar Isi

Contoh OCSF versi 0.1

Berikut ini adalah contoh log menggunakan logging default OCSF versi 0.1.

Contoh-contoh

- Akses diberikan dengan OIDC
- Akses diberikan dengan OIDC dan JAMF
- Akses diberikan dengan OIDC dan CrowdStrike
- Akses ditolak karena cookie hilang
- Akses ditolak oleh kebijakan
- Entri log tidak dikenal

Akses diberikan dengan OIDC

Dalam entri log contoh ini, Akses Terverifikasi memungkinkan akses ke titik akhir dengan penyedia kepercayaan pengguna OIDC.


```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
```

```
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
```

```
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```

Akses diberikan dengan OIDC dan JAMF

Dalam entri log contoh ini, Akses Terverifikasi memungkinkan akses ke titik akhir dengan penyedia kepercayaan perangkat OIDC dan JAMF.

```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0,
        "uid": "41b07859-4222-4f41-f3b9-97dc1EXAMPLE"
    },
    "duration": "0.347",
    "end_time": "1668804944086",
    "time": "1668804944086",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
```

```
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 304
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "oidc",
            "uid": "vatp-9778003bc2EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "4f040d0f96becEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-321318ce-6100d340adf4fb29dEXAMPLE",
        "logged_time": 1668805278555,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-18T20:55:44.086480Z",
    "proxy": {
        "ip": "10.5.192.96",
        "port": 443,
```

```
        "svc_name": "Verified Access",
        "uid": "vai-3598f66575EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "192.168.20.246",
        "port": 61769
    },
    "start_time": "1668804943739",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```

Akses diberikan dengan OIDC dan CrowdStrike

Dalam entri log contoh ini, Akses Terverifikasi memungkinkan akses ke titik akhir dengan OIDC dan penyedia kepercayaan CrowdStrike perangkat.

```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.173.3",
        "os": {
            "name": "Windows 11",
            "type": "Windows",
            "type_id": 100
        },
        "type": "Unknown",
        "type_id": 0,
        "uid": "122978434f65093aee5dfbdc0EXAMPLE",
        "hw_info": {
            "serial_number": "751432a1-d504-fd5e-010d-5ed11EXAMPLE"
```

```
        }
    },
    "duration": "0.028",
    "end_time": "1668816620842",
    "time": "1668816620842",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "test.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://test.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 304
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "oidc",
            "uid": "vatp-506d9753f6EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "23bb45b16a389EXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-c16c5a65-b641e4056cc6cb0eeEXAMPLE",
```

```
        "logged_time": 1668816977134,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-19T00:10:20.842295Z",
    "proxy": {
        "ip": "192.168.144.62",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-2f80f37e64EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.14.173.3",
        "port": 55706
    },
    "start_time": "1668816620814",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```


Akses ditolak karena cookie hilang

Dalam entri log contoh ini, Akses Terverifikasi menolak akses karena cookie otentikasi hilang.


```
{
    "activity": "Access Denied",
    "activity_id": "2",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": null,
    "duration": "0.0",
    "end_time": "1668593568259",
```

```
    "time": "1668593568259",
    "http_request": {
        "http_method": "POST",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/dns-query",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/dns-query"
        },
        "user_agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 302
    },
    "identity": null,
    "message": "",
    "metadata": {
        "uid": "Root=1-5cf1c832-a565309ce20cc7dafEXAMPLE",
        "logged_time": 1668593776720,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T10:12:48.259762Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-108ed7a672EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.7.178.16",
        "port": "46246"
    },
    "start_time": "1668593568258",
    "status_code": "200",
    "status_details": "Authentication Denied",
    "status_id": "2",
```

```
    "status": "Failure",
    "type_uid": "20800102",
    "type_name": "AccessLogs: Access Denied",
    "unmapped": null
}
```

Akses ditolak oleh kebijakan

Dalam entri log contoh ini, Akses Terverifikasi menolak permintaan yang diautentikasi karena permintaan tidak diizinkan oleh kebijakan akses.

```
{
    "activity": "Access Denied",
    "activity_id": "2",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.4.133.137",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.023",
    "end_time": "1668573630978",
    "time": "1668573630978",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 401
    },
    "identity": {
```

```
        "authorizations": [],
        "idp": {
            "name": "user",
            "uid": "vatp-e048b3e0f8EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "0e1281ad3580aEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-531a036a-09e95794c7b96aefbEXAMPLE",
        "logged_time": 1668573773753,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T04:40:30.978732Z",
    "proxy": {
        "ip": "3.223.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-021d5eaed2EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.4.133.137",
        "port": "31746"
    },
    "start_time": "1668573630955",
    "status_code": "300",
    "status_details": "Authorization Denied",
    "status_id": "2",
    "status": "Failure",
    "type_uid": "20800102",
    "type_name": "AccessLogs: Access Denied",
    "unmapped": null
```

```
}
```

Entri log tidak dikenal

Dalam entri log contoh ini, Akses Terverifikasi tidak dapat menghasilkan entri log lengkap sehingga memancarkan entri log yang tidak dikenal. Ini memastikan bahwa setiap permintaan muncul di log akses.

```
{
    "activity": "Unknown",
    "activity_id": "0",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": null,
    "duration": "0.004",
    "end_time": "1668580207898",
    "time": "1668580207898",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "identity": null,
    "message": "",
    "metadata": {
        "uid": "Root=1-435eb955-6b5a1d529343f5adaEXAMPLE",
        "logged_time": 1668580579147,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
```

```
        }
    },
    "ref_time": "2022-11-16T06:30:07.898344Z",
    "proxy": {
        "ip": "10.1.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-6c32b53b3cEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.28.57.68",
        "port": "47220"
    },
    "start_time": "1668580207893",
    "status_code": "000",
    "status_details": "Unknown",
    "status_id": "0",
    "status": "Unknown",
    "type_uid": "20800100",
    "type_name": "AccessLogs: Unknown",
    "unmapped": null
}
```

Contoh OCSF versi 1.0.0-rc.2

Daftar Isi

Akses yang diberikan dengan konteks kepercayaan disertakan

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
```

```
        }],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "invoked_by": "",
        "process": {},
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        },
        "session": {}
    },
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Access Activity",
    "class_uid": "3006",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "message": "",
    "metadata": {
```

```
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_detail": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300601",
    "type_name": "Access Activity: Access Grant",
    "data": {
        "context": {
            "oidc": {
                "family_name": "Last",
                "zoneinfo": "America/Los_Angeles",
                "exp": 1670631145,
                "middle_name": "Middle",
                "given_name": "First",
                "email_verified": true,
                "name": "Test User Display",
                "updated_at": 1666305953,
                "preferred_username": "johndoe-user@test.com",
                "profile": "http://www.example.com",
                "locale": "US",
                "nickname": "Tester",
                "email": "johndoe-user@test.com"
```

```
            },
            "http_request": {
                "x_forwarded_for": "1.1.1.1,2.2.2.2",
                "http_method": "GET",
                "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
                "port": "80",
                "hostname": "hostname.net"
            }
        }
    }
}
```

Akses yang diberikan dengan konteks kepercayaan dihilangkan

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "invoked_by": "",
        "process": {},
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        },
        "session": {}
    },
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Access Activity",
    "class_uid": "3006",
```

```
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
```

```
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_detail": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300601",
    "type_name": "Access Activity: Access Grant",
    "data": null
}
```


Log panggilan API Akses AWS Terverifikasi menggunakan AWS CloudTrail

AWSAkses terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS di Akses. CloudTrailmerekam semua panggilan API untuk Akses Terverifikasi sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari konsol Akses dan panggilan kode ke operasi API Verified Access. Jika membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan CloudTrail peristiwa ke bucket Amazon S3, termasuk peristiwa untuk Akses. Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan olehCloudTrail, Anda dapat menentukan permintaan yang dibuat ke Akses, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjutCloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi Akses Terverifikasi di CloudTrail

CloudTraildiaktifkan pada Anda Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Akses, aktivitas tersebut dicatat dalam CloudTrail peristiwa bersama peristiwa lainnya di Riwayat Layanan AWS peristiwa lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan riwayat CloudTrail peristiwa.

Untuk catatan berkelanjutan tentang peristiwa di AndaAkun AWS, termasuk peristiwa untuk Akses yang berkelanjutan, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke

bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi lainnya Layanan AWS untuk dianalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, lihat yang berikut:

- Gambaran umum untuk membuat jejak
- CloudTrailLayanan yang didukung dan integrasi
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file log CloudTrail dari beberapa wilayah dan Menerima file log CloudTrail dari beberapa akun

Semua tindakan Akses dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API Amazon EC2. Misalnya, panggilan untuk tindakan `CreateVerifiedAccessInstance`, `DeleteVerifiedAccessInstance`, dan `ModifyVerifiedAccessInstance` menghasilkan entri di berkas log CloudTrail.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan kredensi pengguna root atau AWS Identity and Access Management (IAM).
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh orang lainLayanan AWS.

Untuk informasi selengkapnya, lihat Elemen CloudTrail userIdentity.

Memahami entri berkas log Akses

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang telah Anda tentukan. File log CloudTrail berisi satu atau lebih entri log. Suatu peristiwa mewakili permintaan tunggal dari semua sumber. Peristiwa ini mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Berkas log CloudTrail bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log untuk `CreateVerifiedAccessInstance` tindakan.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIKK4OOINJWEXAMPLE:jdoe",
        "arn": "arn:aws:iam::123456789012:user/jdoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "jdoe"
    },
    "eventTime": "2022-11-18T20:44:04Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateVerifiedAccessInstance",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "CreateVerifiedAccessInstanceRequest": {
            "Description": "",
            "ClientToken": "85893b1e-49f6-4d24-97de-280c664edf1b"
        }
    },
    "responseElements": {
        "CreateVerifiedAccessInstanceResponse": {
            "verifiedAccessInstance": {
                "creationTime": "2022-11-18T20:44:04",
                "description": "",
                "verifiedAccessInstanceId": "vai-0d79d91875542c549",
                "verifiedAccessTrustProviderSet": ""
            },
            "requestId": "2eae195d-6bfd-46d7-b46e-a68cb791de09"
        }
    },
    "requestID": "2eae195d-6bfd-46d7-b46e-a68cb791de09",
    "eventID": "297d6529-1144-40f6-abf8-3a76f18d88f0",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
```

```
}
```

Kuota untuk Akses AWS Terverifikasi

Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk masing-masing. Layanan AWS Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah.

Akun AWS-kuota tingkat

Anda Akun AWS memiliki kuota berikut yang terkait dengan Akses Terverifikasi.

Nama	Default	Dapat Disesuaikan	Deskripsi
Instans Akses Terverifikasi	5	Ya	Jumlah maksimum Instans Akses Terverifikasi yang dapat dibuat pelanggan di Wilayah saat ini.
Grup Akses Terverifikasi	10	Ya	Jumlah maksimum Grup Akses Terverifikasi yang dapat dibuat pelanggan di Wilayah saat ini.
Penyedia Kepercayaan Akses Terverifikasi	15	Ya	Jumlah maksimum Penyedia Trust Akses Terverifikasi yang dapat dibuat pelanggan di Wilayah saat ini.
Titik Akhir Akses Terverifikasi	50	Ya	Jumlah maksimum Titik Akhir Akses Terverifikasi yang dapat dibuat pelanggan di Wilayah saat ini.

Header HTTP

Berikut ini adalah batas ukuran untuk header HTTP.

Nama	Default	Dapat Disesuaikan
Baris permintaan	16 K	Tidak
Header tunggal	16 K	Tidak

Nama	Default	Dapat Disesuaikan
Seluruh header respon	32 K	Tidak
Seluruh header permintaan	64 K	Tidak

Ukuran klaim OIDC

Berikut ini adalah batas ukuran klaim OIDC.

Nama	Default	Dapat Disesuaikan
Ukuran klaim OIDC	11 K	Tidak

Riwayat dokumen untuk Panduan Pengguna Akses Terverifikasi

Tabel berikut menjelaskan rilis dokumentasi untuk Akses Terverifikasi.

Perubahan	Deskripsi	Tanggal
AWSkebijakan terkelola diperbarui	Pembaruan dibuat untuk kebijakan IAM AWS terkelola untuk Akses Terverifikasi.	17 November 2023
Enkripsi data saat istirahat	AWSAkses Terverifikasi mengenkripsi data saat istirahat secara default, menggunakan kunci KMS yang AWS dimiliki.	September 28, 2023
Support untuk kepatuhan FIPS	Konfigurasikan Akses Terverifikasi untuk kepatuhan FIPS.	26 September 2023
Penebangan yang ditingkatkan	Penambahan fitur logging yang menambahkan konteks kepercayaan ke log.	19 Juni 2023
AWSkebijakan terkelola diperbarui	Pembaruan dibuat untuk kebijakan IAM AWS terkelola untuk Akses Terverifikasi.	31 Mei 2023
Rilis GA	Rilis GA dari Panduan Pengguna Akses Terverifikasi. Termasuk AWS WAFintegrasi.	April 27, 2023
Rilis pratinjau	Pratinjau rilis Panduan Pengguna Akses Terverifikasi	29 November 2022

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.