Membangun pagar pembatas dan pemantauan untuk URL yang telah ditetapkan sebelumnya

AWS Panduan Preskriptif

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

AWS Panduan Preskriptif: Membangun pagar pembatas dan pemantauan untuk URL yang telah ditetapkan sebelumnya

Copyright © 2026 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 pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Membangun pagar pembatas dan pemantauan untuk presigned URLs

Ryan Baker, Amazon Web Services ()AWS

Agustus 2025 (sejarah dokumen)

Keamanan adalah perhatian penting bagi semua perusahaan dan pilar utama dalam AWS Well-Architected Framework. Sebagai insinyur keamanan, Anda akan ingin menerapkan pagar pembatas administratif yang selaras dengan persyaratan kontrol organisasi. Dalam AWS Well-Architected Framework, pagar pembatas menentukan batas-batas yang membatasi aktivitas.

Panduan ini memberikan informasi latar belakang dan praktik terbaik untuk menggunakan presigned URLs, yang digunakan dengan objek Amazon Simple Storage Service (Amazon S3). Presigned URLs memungkinkan pengguna atau aplikasi yang memiliki akses ke kredenial yang valid untuk menghasilkan permintaan yang ditandatangani sebelumnya dan diterima hingga waktu kedaluwarsa yang ditentukan. Kasus penggunaan umum untuk presigned URLs adalah memperluas akses ke objek atau sumber daya dengan membagikan permintaan ini. Permintaan presigned bersama dihasilkan oleh sistem atau pengguna yang memiliki hak untuk melakukan permintaan tertentu, dan kemudian dapat dikirim ke sistem atau pengguna lain untuk memperluas kemampuan untuk melakukan permintaan yang sama.

Dalam panduan ini, Anda akan belajar:

- Konsep presigned URLs
- Gunakan kasus untuk presigned URLs
- Pagar pembatas yang direkomendasikan dan opsional
- Opsi pemantauan
- Contoh bagaimana Layanan AWS penggunaan presigned URLs

Audiens yang dituju

Panduan ini ditujukan untuk arsitek dan insinyur keamanan yang bertanggung jawab untuk menerapkan kontrol keamanan di AWS Cloud.

Tujuan

Sebagai insinyur keamanan, Anda ingin mengetahui bagaimana pembuat solusi menerapkan keamanan dan jenis akses yang dimiliki pengguna akhir Anda. Panduan ini mencakup satu jenis akses, presigned URLs, yang sering digunakan dengan Amazon S3. Presigned URLs memberi pembangun opsi untuk menjembatani mekanisme otentikasi secara efisien.

Di Amazon S3, presigned URLs mewakili kategori permintaan yang unik. Insinyur keamanan dapat memantau dan mengelola permintaan ini untuk memastikan bahwa mereka hanya digunakan jika sesuai dan perlu. Tujuan dari panduan ini adalah untuk membantu insinyur keamanan memberikan jenis pengawasan tingkat tinggi ini.

Setelah membaca panduan ini, Anda harus memahami apa itu URL yang telah ditentukan sebelumnya, kapan biasanya digunakan, dan motivasi penggunaannya.

Prasyarat

Jika perusahaan Anda belum menetapkan kebijakan keamanan, tujuan kontrol, atau standar, seperti yang dijelaskan dalam panduan Menerapkan kontrol keamanan di AWS, sebaiknya Anda menyelesaikan tugas tata kelola tersebut sebelum melanjutkan dengan panduan ini.

Sebelum memulai, Anda juga harus terbiasa dengan praktik terbaik yang direkomendasikan dan opsional untuk kontrol dan pemantauan. Untuk informasi lebih lanjut, lihat:

- Kebijakan kontrol layanan (AWS Organizations dokumentasi)
- Kebijakan kontrol sumber daya (AWS Organizations dokumentasi)
- Kebijakan bucket untuk Amazon S3 (dokumentasi Amazon S3)
- Permintaan logging dengan pencatatan akses server (dokumentasi Amazon S3)
- Pencatatan panggilan API Amazon S3 menggunakan AWS CloudTrail(dokumentasi Amazon S3)

Ikhtisar presigned URLs

URL presigned adalah jenis permintaan HTTP yang dikenali oleh layanan AWS Identity and Access Management (IAM). Apa yang membedakan jenis permintaan ini dari semua AWS permintaan lainnya adalah parameter X-Amz-Expires kueri. Seperti permintaan otentikasi lainnya, permintaan URL presigned menyertakan tanda tangan. Untuk permintaan URL yang telah ditetapkan sebelumnya, tanda tangan ini dikirimkan`X-Amz-Signature`. Tanda tangan menggunakan operasi kriptografi Signature Version 4 untuk menyandikan semua parameter permintaan lainnya.

> **Catatan**
>
> - Signature Version 2 saat ini sedang dalam proses tidak digunakan lagi, tetapi masih didukung di beberapa. Wilayah AWS Panduan ini berlaku untuk penandatanganan Signature Version 4.
> - Layanan penerima dapat memproses header yang tidak ditandatangani, tetapi dukungan untuk opsi itu terbatas dan ditargetkan, sejalan dengan praktik terbaik. Kecuali dinyatakan lain, asumsikan bahwa semua header harus ditandatangani agar permintaan diterima.

`X-Amz-Expires`Parameter memungkinkan tanda tangan diproses sebagai valid dengan penyimpangan yang lebih besar dari waktu tanggal yang dikodekan. Aspek lain dari validitas tanda tangan masih dievaluasi. Kredensi penandatanganan, jika sementara, tidak boleh kedaluwarsa pada saat tanda tangan diproses. Kredensi penandatanganan harus dilampirkan pada kepala sekolah IAM yang memiliki otorisasi yang memadai pada saat pemrosesan.

Presigned URLs adalah subset dari permintaan presigned

URL presigned bukan satu-satunya metode untuk menandatangani permintaan untuk masa depan. Amazon S3 juga mendukung permintaan POST, yang biasanya juga sudah ditentukan sebelumnya. Tanda tangan POST yang telah ditetapkan sebelumnya memungkinkan unggahan yang mematuhi kebijakan yang ditandatangani dan memiliki tanggal kedaluwarsa yang disematkan dalam kebijakan tersebut.

Tanda tangan untuk permintaan dapat diberi tanggal di masa mendatang, meskipun ini jarang terjadi. Selama kredensi yang mendasarinya valid, algoritma tanda tangan tidak melarang penanggalan masa depan. Namun, permintaan ini tidak dapat berhasil diproses sampai jendela waktu yang valid, yang membuat kencan future tidak praktis untuk sebagian besar kasus penggunaan.

Apa yang diizinkan oleh permintaan yang telah ditetapkan sebelumnya?

Permintaan presigned hanya dapat mengizinkan tindakan yang diizinkan oleh kredensil yang digunakan untuk menandatangani permintaan. Jika kredensil secara implisit atau eksplisit menolak tindakan yang ditentukan oleh permintaan yang telah ditetapkan sebelumnya, permintaan yang telah ditetapkan sebelumnya akan ditolak saat dikirim. Ini berlaku untuk yang berikut:

- Kebijakan sesi yang terkait dengan kredensialnya
- Kebijakan berbasis identitas yang terkait dengan prinsipal yang terkait dengan kredensialnya
- Kebijakan sumber daya yang memengaruhi sesi atau prinsipal
- Kebijakan pengendalian layanan yang memengaruhi sesi atau prinsipal
- Kebijakan pengendalian sumber daya yang memengaruhi sesi atau prinsipal

Motivasi untuk menggunakan permintaan yang telah ditentukan sebelumnya

Sebagai insinyur keamanan, Anda harus menyadari apa yang memotivasi pembangun solusi untuk menggunakan URLs presigned. Memahami apa yang diperlukan dan apa yang opsional akan membantu Anda berkomunikasi dengan pembuat solusi. Motivasi mungkin termasuk yang berikut:

- Untuk mendukung mekanisme otentikasi non-IAM sambil memanfaatkan skalabilitas di Amazon S3. Motivasi inti adalah berkomunikasi langsung dengan Amazon S3 untuk mendapatkan manfaat dari skalabilitas bawaan yang disediakan oleh layanan ini. Tanpa komunikasi langsung ini, solusi perlu mendukung beban dari transmisi ulang byte yang dikirim dan panggilan. PutObjectGetObject Bergantung pada beban total, persyaratan ini menambahkan tantangan penskalaan yang mungkin ingin dihindari oleh pembuat solusi.

  Cara lain untuk berkomunikasi langsung dengan Amazon S3, seperti menggunakan kredensi sementara AWS Security Token Service di AWS STS() atau tanda tangan Versi Tanda Tangan 4 di URLs luar, mungkin tidak sesuai untuk kasus penggunaan Anda. Amazon S3 mengidentifikasi pengguna melalui AWS kredensil, sedangkan permintaan yang telah ditetapkan sebelumnya menganggap identifikasi melalui mekanisme selain kredensil. AWS Menjembatani perbedaan ini sambil mempertahankan komunikasi langsung untuk data dapat dicapai melalui permintaan yang telah ditentukan sebelumnya.
- Untuk mendapatkan manfaat dari pemahaman asli browser URLs. URLs dipahami oleh browser, sedangkan AWS STS kredensi dan tanda tangan Versi Tanda Tangan 4 tidak. Ini bermanfaat saat

berintegrasi dengan solusi berbasis browser. Solusi alternatif memerlukan lebih banyak kode, akan menggunakan lebih banyak memori untuk file besar, dan mungkin diperlakukan berbeda dengan ekstensi seperti malware dan pemindai virus.

Perbandingan dengan AWS STS kredensial sementara

Kredensi sementara mirip dengan permintaan yang telah ditetapkan sebelumnya. Keduanya kedaluwarsa, memungkinkan pelingkupan akses, dan biasanya digunakan untuk menjembatani kredenal non-IAM dengan penggunaan yang memerlukan kredensil AWS.

Anda dapat dengan ketat mencakup AWS STS kredensi sementara ke objek dan tindakan S3 tunggal, tetapi ini dapat menghasilkan tantangan penskalaan karena AWS STS APIs memiliki batasan. (Untuk informasi selengkapnya, lihat artikel Bagaimana cara mengatasi pelambatan API atau kesalahan “Nilai terlampaui” untuk IAM dan di situs web AWS AWS STS re:Post.) Selain itu, setiap kredensi yang dihasilkan memerlukan panggilan AWS STS API, yang menambahkan latensi dan ketergantungan baru yang dapat memengaruhi ketahanan. AWS STS Kredensi sementara juga memiliki waktu kedaluwarsa minimum 15 menit, sedangkan permintaan yang telah ditentukan sebelumnya dapat mendukung durasi yang lebih pendek. (60 detik praktis mengingat kondisi yang tepat.)

Perbandingan dengan solusi khusus tanda tangan

Satu-satunya komponen rahasia inheren dari permintaan yang telah ditetapkan sebelumnya adalah tanda tangan Versi Tanda Tangan 4. Jika klien mengetahui detail lain dari permintaan dan diberikan tanda tangan yang valid yang cocok dengan detail tersebut, ia dapat mengirim permintaan yang valid. Tanpa tanda tangan yang valid, itu tidak bisa.

Solusi yang ditetapkan sebelumnya URLs dan hanya tanda tangan serupa secara kriptografis. Namun, solusi khusus tanda tangan memiliki keuntungan praktis seperti kemampuan untuk menggunakan header HTTP alih-alih parameter string kueri untuk mengirimkan tanda tangan (lihat bagian Interaksi dan mitigasi Logging). Administrator juga harus mempertimbangkan bahwa string kueri lebih sering diperlakukan sebagai metadata, sedangkan header lebih jarang diperlakukan seperti itu.

Di sisi lain, AWS SDKs berikan lebih sedikit dukungan untuk menghasilkan dan menggunakan tanda tangan secara langsung. Membangun solusi khusus tanda tangan memerlukan lebih banyak kode khusus. Dari perspektif praktis, menggunakan pustaka alih-alih kode khusus untuk keamanan adalah

praktik terbaik umum, sehingga kode untuk solusi khusus tanda tangan memerlukan pengawasan ekstra.

Solusi khusus tanda tangan tidak menggunakan string `X-Amz-Expires` kueri dan tidak memberikan periode validitas eksplisit. IAM mengelola periode validitas implisit tanda tangan yang tidak memiliki waktu kedaluwarsa eksplisit. Periode implisit tersebut tidak dipublikasikan. Mereka biasanya tidak berubah, tetapi dikelola dengan mempertimbangkan keamanan, jadi Anda tidak boleh bergantung pada periode validitas. Ada tradeoff antara memiliki kontrol eksplisit atas tanggal kedaluwarsa dan meminta IAM mengelola kedaluwarsa.

Sebagai administrator, Anda mungkin lebih memilih solusi khusus tanda tangan. Namun, dalam arti praktis, Anda harus mendukung solusi seperti yang dibangun.

Mengidentifikasi permintaan yang telah ditetapkan sebelumnya

Mengidentifikasi permintaan yang menggunakan URL presigned

Amazon S3 menyediakan dua mekanisme bawaan untuk memantau penggunaan pada tingkat permintaan: log AWS CloudTrail akses server Amazon S3 dan peristiwa data. Kedua mekanisme tersebut dapat mengidentifikasi penggunaan URL yang telah ditentukan sebelumnya.

Untuk memfilter log untuk penggunaan URL yang telah ditetapkan sebelumnya, Anda dapat menggunakan jenis otentikasi. Untuk log akses server, periksa bidang Jenis Otentikasi, yang biasanya diberi nama authtype saat didefinisikan dalam tabel Amazon Athena. Untuk CloudTrail, periksa AuthenticationMethoddi `additionalEventData` lapangan. Dalam kedua kasus, nilai bidang untuk permintaan yang menggunakan URL presigned adalah`QueryString`, sedangkan `AuthHeader` nilai untuk sebagian besar permintaan lainnya.

`QueryString`penggunaan tidak selalu dikaitkan dengan URL yang telah ditetapkan sebelumnya. Untuk membatasi penelusuran Anda hanya menggunakan URL yang telah ditentukan sebelumnya, temukan permintaan yang berisi parameter string kueri. `X-Amz-Expires` Untuk log akses server, periksa request-URI dan cari permintaan yang memiliki `X-Amz-Expires` parameter dalam string kueri. Untuk CloudTrail, periksa `requestParameters` elemen untuk `X-Amz-Expires` elemen.

```
{"Records": [{…, "requestParameters": {…, "X-Amz-Expires": "300"}}, …]}
```

Kueri Athena berikut menerapkan filter ini:

```
SELECT * FROM {athena-table} WHERE
  authtype = 'QueryString' AND
  request_uri LIKE '%X-Amz-Expires=%';
```

Untuk AWS CloudTrail Lake, kueri berikut menerapkan filter ini:

```
SELECT * FROM {data-store-event-id} WHERE
  additionalEventData['AuthenticationMethod'] = 'QueryString' AND
  requestParameters['X-Amz-Expires'] IS NOT NULL
```

Mengidentifikasi jenis permintaan presigned lainnya

Permintaan POST juga memiliki jenis otentikasi unik,`HtmlForm`, di log akses server Amazon S3 dan. CloudTrail Jenis otentikasi ini kurang umum, jadi Anda mungkin tidak menemukan permintaan ini di lingkungan Anda.

Kueri Athena berikut menerapkan filter untuk: `HtmlForm`

```
SELECT * FROM {athena-table} WHERE
  authtype = 'HtmlForm';
```

Untuk CloudTrail Lake, kueri berikut menerapkan filter:

```
SELECT * FROM {data-store-event-id} WHERE
  additionalEventData['AuthenticationMethod'] = 'HtmlForm'
```

Mengidentifikasi pola permintaan

Anda dapat menemukan permintaan yang telah ditentukan sebelumnya dengan menggunakan teknik yang dibahas di bagian sebelumnya. Namun, untuk membuat data itu berguna, Anda akan ingin menemukan pola. `TOP 10`Hasil sederhana untuk kueri Anda mungkin memberikan wawasan, tetapi jika itu tidak cukup, gunakan opsi pengelompokan dalam tabel berikut.

Opsi pengelompokan	Log akses server	CloudTrailDanau	Deskripsi
Agen pengguna	`GROUP BY useragent`	`GROUP BY userAgent`	Opsi pengelompokan ini membantu Anda menemukan sumber dan tujuan permintaan. Agen pengguna disediakan oleh pengguna dan tidak dapat diandalkan sebagai mekanisme otentikasi atau otorisasi. Namun, ini dapat

Opsi pengelompokan	Log akses server	CloudTrailDanau	Deskripsi
			mengungkapkan banyak hal jika Anda mencari pola, karena sebagian besar klien menggunakan string unik yang setidakny a sebagian dapat dibaca manusia.

Opsi pengelompokan	Log akses server	CloudTrailDanau	Deskripsi
Pemohon	`GROUP BY requester`	`GROUP BY userIdentity['arn']`	Opsi pengelompokan ini membantu menemukan kepala sekolah IAM yang menandatangani permintaan. Jika tujuan Anda adalah untuk memblokir permintaan ini atau membuat pengecualian untuk permintaan yang ada, kueri ini memberikan informasi yang cukup untuk tujuan itu. Ketika Anda menggunakan peran sesuai dengan praktik terbaik IAM, peran tersebut memiliki pemilik yang diidentifikasi dengan jelas, dan Anda dapat menggunakan informasi tersebut untuk mengetahui lebih lanjut.

Opsi pengelompokan	Log akses server	CloudTrailDanau	Deskripsi
Alamat IP sumber	`GROUP BY remoteip`	`GROUP BY sourceIPAddress`	Opsi ini dikelompokkan berdasarkan lompatan terjemahan jaringan terakhir sebelum mencapai Amazon S3. • Jika lalu lintas melewati gateway NAT, ini akan menjadi alamat NAT Gateway. • Jika lalu lintas melewati gateway internet, ini akan menjadi alamat IP publik yang mengirim lalu lintas ke gateway internet. • Jika lalu lintas berasal dari luar AWS, ini akan menjadi alamat internet publik yang terkait dengan asal. • Jika melewati titik akhir gateway virtual private cloud (VPC), ini akan

Opsi pengelompokan	Log akses server	CloudTrailDanau	Deskripsi
			menjadi alamat IP instance di VPC. • Jika melewati antarmuka virtual publik (VIF), ini akan menjadi IP lokal pemohon atau perantara apa pun seperti server proxy atau firewall yang hanya mengekspos alamat IP-nya. • Jika melewati titik akhir VPC antarmuka, ini bisa menjadi alamat IP dari sebuah instance di VPC. Ini juga bisa berupa alamat IP dari VPC lain atau jaringan lokal. Seperti halnya VIF publik, ini bisa berupa alamat IP perantara apa pun. Data ini berguna jika tujuan Anda adalah memaksaka n kontrol jaringan.

Opsi pengelompokan	Log akses server	CloudTrailDanau	Deskripsi
			Anda mungkin harus menggabungkan opsi ini dengan data seperti `endpoint` (untuk log akses server) atau `vpcEndpointId` (untuk CloudTrail Lake) untuk memperjelas sumbernya, karena jaringan yang berbeda mungkin menduplikasi alamat IP pribadi.
Nama ember S3	`GROUP BY bucket_name`	`GROUP BY requestParameters['bucketName']`	Opsi pengelompokan ini membantu menemukan bucket yang menerima permintaan. Ini membantu Anda mengidentifikasi kebutuhan akan pengecualian.

Praktik terbaik untuk menggunakan permintaan yang telah ditetapkan sebelumnya

Bagian ini membahas praktik terbaik untuk menggunakan permintaan yang telah ditetapkan sebelumnya yang harus dipertimbangkan oleh insinyur keamanan. Pedoman tersebut meliputi:

- Praktik terbaik dasar, yang merupakan praktik yang harus diikuti oleh setiap organisasi.
- Pagar pembatas tambahan, yang merupakan praktik yang harus Anda pertimbangkan, tetapi mungkin memutuskan untuk menerapkan sebagian atau dengan pengecualian. Ini dimaksudkan untuk memberikan kontrol dan pertahanan tambahan secara mendalam, tetapi harus diimbangi dengan kompleksitas keseluruhan.
- Interaksi logging, yang mungkin dihasilkan dari perangkat atau layanan yang merupakan bagian dari tanggung jawab Anda atau pelanggan Anda dalam model tanggung jawab bersama. Bagian ini mencakup tindakan pencegahan untuk membatasi informasi yang dapat diakses melalui log.

Praktik terbaik dasar

Praktik terbaik umum yang merupakan kontrol efektif untuk permintaan AWS API lainnya juga berlaku untuk permintaan yang telah ditetapkan sebelumnya. Bagian ini mengulas dua praktik yang paling relevan: hak istimewa terkecil dan batas data. Praktik-praktik ini menciptakan kedalaman kontrol yang diperluas oleh praktik lain.

Terapkan prinsip hak istimewa paling sedikit

Langkah pertama dalam membatasi penggunaan permintaan presigned adalah membatasi akses ke Amazon S3 secara umum. URL presigned tidak dapat menyediakan akses ke sumber daya yang tidak diberikan kepada prinsipal yang menghasilkan tanda tangan untuk URL yang telah ditetapkan sebelumnya. Juga tidak dapat menyediakan akses ke sumber daya dengan cara yang tidak diberikan kepada kepala sekolah itu. Dengan demikian, menerapkan praktik terbaik untuk memberikan hak istimewa paling sedikit kepada para prinsipal tersebut adalah pagar pembatas yang efektif.

Proses pembuatan URL presigned adalah operasi algoritmik yang didasarkan pada standar yang diterbitkan (Signature Version 4) untuk pembuatan tanda tangan. Oleh karena itu, tidak mungkin untuk membatasi pembuatan URL yang telah ditentukan sebelumnya. Namun, agar relevan, URL

yang telah ditetapkan sebelumnya harus valid dan menyediakan akses ke sumber daya, sehingga validitas URL yang telah ditetapkan sebelumnya juga merupakan pagar pembatas yang efektif.

Untuk informasi selengkapnya tentang hak istimewa terkecil, lihat Berikan akses hak istimewa paling sedikit di pilar AWS Well-Architected Kerangka, Keamanan.

Menerapkan perimeter data

Perpanjangan hak istimewa terkecil adalah mempertahankan perimeter data yang konsisten dengan kebutuhan organisasi Anda. URL presigned kompatibel dengan perimeter data. Seperti permintaan lainnya, validitas permintaan URL presigned dievaluasi pada waktu permintaan. Jika properti jaringan, sumber daya, sesi peran, dan prinsipal berubah, mereka dievaluasi pada saat itu dan dengan menggunakan metode dimana permintaan diterima.

Misalnya, katakanlah layanan yang berjalan di container Amazon Elastic Kubernetes Service (Amazon EKS) menandatangani permintaan. Permintaan tersebut kemudian dikirim dari sistem komputer pribadi pengguna yang terhubung ke internet. Dalam hal ini, SourceIp kondisi aws: mengevaluasi alamat IP publik yang terlihat dari permintaan dari sistem pribadi pengguna, bukan alamat IP layanan di wadah Amazon EKS.

Demikian pula, jika tag prinsipal atau sumber daya berubah sebelum permintaan dikirim, nilai yang diperbarui, bukan asli, akan berlaku untuk permintaan melalui ResourceTag/tag-key kondisi aws: PrincipalTag/tag-key dan aws:.

Pagar pembatas tambahan

Ketika permintaan presigned digunakan dengan tepat oleh pembuat solusi dan pengguna, mereka menyediakan mekanisme yang aman untuk memberikan pengguna akses ke data. Selain itu, kemampuan untuk menghasilkan permintaan yang telah ditetapkan sebelumnya tidak memberikan akses kepada prinsipal yang belum mereka miliki.

Dalam konteks itu, apakah kontrol tambahan diperlukan? Pembenaran untuk kontrol tambahan tidak didasarkan pada kebutuhan untuk menolak akses tetapi untuk menyediakan kemampuan untuk memantau, untuk menyetujui penggunaan dan menetapkan batasan, dan untuk mengurangi risiko dari kesalahan pengguna. Dengan cara ini Anda dapat membantu memastikan bahwa penggunaan sesuai dan perlu.

Pagar pembatas berikut membantu Anda dalam tujuan ini. Sebelum mengaktifkan kontrol ini, Anda mungkin ingin menentukan penggunaan yang ada dengan mengidentifikasi permintaan yang telah

ditetapkan sebelumnya. Identifikasi ini membantu Anda mempersiapkan dampak pagar pembatas terhadap penggunaan yang ada atau merencanakan pengecualian jika diperlukan.

Pagar pembatas untuk S3: SignatureAge

Salah satu karakteristik yang menentukan dari permintaan yang telah ditetapkan sebelumnya adalah bahwa mereka menggambarkan waktu kedaluwarsa. Tanda tangan untuk permintaan berisi tanggal. Tanggal ini ditransmisikan sebagai parameter string `X-Amz-Date` kueri untuk URL yang telah ditetapkan sebelumnya, dan sebagai header Tanggal atau x-amz-date untuk POST yang telah ditetapkan sebelumnya.

Amazon S3 menyediakan kunci kondisi, S3:SignatureAge, yang dapat Anda gunakan untuk membatasi waktu maksimum antara tanggal yang ditandatangani dan kedaluwarsa efektif permintaan. Kondisi ini tidak pernah dapat meningkatkan masa berlaku, tetapi dapat menguranginya.

Dalam kebijakan berikut, kunci `s3:signatureAge` kondisi membatasi permintaan yang telah ditetapkan sebelumnya hingga 15 menit validitas. Contoh berikut semua menggunakan 15 menit untuk membatasi validitas untuk jangka waktu yang sama sebagai dukungan penandatanganan standar.

Pernyataan kedua dari kebijakan tersebut menolak akses Signature Version 2. Versi protokol penandatanganan ini tidak digunakan lagi, tetapi masih didukung di beberapa. Wilayah AWS Kami menyarankan Anda memblokirnya secara eksplisit sebelum sepenuhnya tidak digunakan lagi.

Anda dapat menerapkan kebijakan berikut sebagai kebijakan kontrol AWS Organizations layanan (SCP). Pengguna masih dapat menggunakan permintaan yang telah ditetapkan sebelumnya dan menerapkan solusi yang bergantung pada permintaan tersebut, selama waktu antara pembuatan tanda tangan dan penggunaan kurang dari 15 menit. Bergantung pada implementasinya, batasan ini mungkin tidak berdampak, dapat menyebabkan solusi menjadi tidak dapat digunakan, atau mungkin menyebabkan kegagalan sesekali yang dapat dicoba ulang.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15Minutes",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
```

```
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        }
      }
    },
    {
      "Sid": "DenySignatureVersion2",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
           "s3:signatureversion": "AWS"
        }
      }
    }
  ]
}
```

Pengecualian

Jika solusi memerlukan waktu yang lebih lama sebelum kedaluwarsa dan oleh karena itu dipengaruhi oleh kebijakan sebelumnya, kami sarankan Anda menyediakan metode untuk menyetujui pengecualian. Untuk menghindari penghitungan pengecualian dalam SCP, gunakan aws:, seperti dalam kebijakan berikutPrincipalTag, untuk mengelola pengecualian dengan cara yang dapat diskalakan. AWS Contoh lain, seperti contoh kebijakan perimeter data AWS, menggunakan strategi ini.

Jika Anda menerapkan kebijakan pengecualian dengan menggunakan`aws:PrincipalTag`, Anda harus mengontrol akses ke pengaturan tag pada prinsipal. Tag jenis ini dapat datang langsung dari prinsipal dan dapat dikontrol oleh SCP, seperti dalam contoh pengendalian nilai tag mana yang dapat diatur. Tag jenis ini juga dapat berasal dari tag sesi, yang ditetapkan oleh penyedia identitas (iDP) atau saat menggunakan. AWS STS Mengontrol akses ke `aws:PrincipalTag` adalah topik yang kompleks. Namun, organisasi yang memiliki pengalaman dalam menggunakan kontrol akses berbasis atribut (ABAC) akan memiliki pengalaman dan kontrol untuk memungkinkan penggunaan yang tepat `aws:PrincipalTag` untuk kasus penggunaan ini.

Dalam contoh berikut, `aws:PrincipalTag` kondisi membuat pengecualian yang memungkinkan prinsipal apa pun dengan tag bernama (`long-presigned-allowed`) ditetapkan dan disetel ke`true`. Dengan pengecualian ini pembatasan usia tanda tangan tidak diterapkan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15Minutes",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/long-presigned-allowed": "true"
        }
      }
    }
  ]
}
```

Kebijakan bucket

Anda dapat menerapkan kebijakan bucket ke semua atau bucket yang dipilih dengan menggunakan kebijakan seperti pada contoh berikut. Tidak seperti SCP, kebijakan bucket juga menargetkan penggunaan utama layanan. Lampiran A tidak mendokumentasikan penggunaan utama layanan yang diharapkan dari permintaan yang telah ditetapkan sebelumnya, tetapi jika Anda ingin menerapkan kontrol untuk membuktikan batas tersebut, kebijakan berikut akan memberikan kontrol tersebut. Selain itu, tidak seperti SCP, kebijakan bucket dapat berlaku untuk prinsipal di akun manajemen Anda.

ABAC-based pengecualian bekerja dalam kebijakan bucket dengan cara yang sama seperti SCP. Tujuan dari kebijakan bucket mungkin untuk diterapkan pada prinsipal di luar organisasi, sehingga pengecualian ABAC harus dibatasi pada prinsip-prinsip yang berlaku kontrol ABAC.

Dalam contoh berikut, `aws:PrincipalTag` kondisi dalam pernyataan pertama membuat pengecualian untuk prinsipal dengan tag bernama (`long-presigned-allowed`) ditetapkan dan disetel `ketrue`. Dengan pengecualian ini pembatasan usia tanda tangan tidak diterapkan. Pernyataan kedua menerapkan pembatasan ini untuk semua kepala sekolah di luar AWS organisasi yang memiliki ember. Ruang lingkup pernyataan kedua ini harus cocok dengan kontrol ABAC untuk mengatur tag bernama untuk prinsipal.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15MinWithExceptions",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/long-presigned-allowed": "true"
        }
      }
    },
    {
      "Sid": "DenyPresignedOver15MinutesOutsideOrg",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalOrgID": "${aws:ResourceOrgID}"
        }
      }
    }
  ]
}
```

Kebijakan kontrol sumber daya

Anda dapat menerapkan kebijakan ke bucket dalam skala besar dengan menggunakan kebijakan kontrol sumber daya (RCP). Seperti SCP dan tidak seperti kebijakan bucket, RCP tidak menargetkan penggunaan utama layanan. RCP memengaruhi prinsip non-layanan dari akun apa pun, tetapi

tidak memengaruhi sumber daya di akun manajemen. Lihat informasi yang lebih lengkap dalam dokumentasi AWS Organizations.

Seperti kebijakan bucket, jika Anda gunakan `aws:PrincipalTags` untuk membuat pengecualian untuk prinsipal, perhatikan cakupan kontrol ABAC pada penandaan prinsipal.

RCP berikut membatasi penggunaan URL yang telah ditetapkan sebelumnya di semua bucket S3 dalam organisasi dengan membatasi usia tanda tangan hingga 15 menit.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPresignedOver15MinWithExceptions",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/long-presigned-allowed": "true",
        }
      }
    },
    {
      "Sid": "DenyPresignedOver15MinutesOutsideOrg",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:signatureAge": "900000"
        },
        "StringNotEquals": {
          "aws:PrincipalOrgID": "${aws:ResourceOrgID}"
        }
      }
    }
  ]
```

```
}
```

Guardrail untuk S3:AuthType

URL presigned menggunakan otentikasi string kueri, dan POST presigned selalu menggunakan otentikasi POST. Amazon S3 mendukung penolakan permintaan berdasarkan jenis otentikasi melalui kunci kondisi S3:authType. `REST-QUERY-STRING`adalah `s3:authType` nilai untuk string kueri, dan `POST` merupakan `s3:authType` nilai untuk POST.

Anda dapat menerapkan kebijakan berikut sebagai SCP. Kebijakan ini digunakan `s3:authType` untuk mengizinkan hanya autentikasi berbasis header. Ini juga mengkonfigurasi metode untuk memberikan pengecualian kepada pengguna individu atau peran.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyNonHeaderAuth",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "s3:authType": "REST-HEADER",
          "aws:PrincipalTag/non-header-auth-allowed": "true"
        }
      }
    }
  ]
}
```

Menolak permintaan berdasarkan jenis otentikasi memengaruhi solusi atau fitur apa pun yang menggunakan jenis otentikasi yang ditolak. Misalnya, menyangkal `REST-QUERY-STRING` mencegah pengguna melakukan unggahan atau unduhan dari konsol Amazon S3. Jika Anda ingin pengguna menggunakan konsol Amazon S3, jangan gunakan pagar pembatas ini, atau buat pengecualian untuk pengguna. Di sisi lain, jika Anda tidak ingin pengguna menggunakan konsol Amazon S3, Anda dapat menolak `REST-QUERY-STRING` untuk pengguna.

Mungkin Anda sudah menolak akses langsung pengguna ke sumber daya Amazon S3. Dalam hal ini, pagar pembatas untuk jenis otentikasi berlebihan. Namun, pernyataan penolakan `s3:authType`

menyediakan utilitas defense-in-depth karena implementasi untuk menolak akses langsung biasanya mencakup banyak pernyataan kontrol, beberapa dengan pengecualian.

Peran yang digunakan untuk beban kerja biasanya tidak memerlukan akses ke string kueri atau `POST` otentikasi. Pengecualian adalah peran yang mendukung layanan yang dirancang untuk menggunakan permintaan yang telah ditetapkan sebelumnya. Anda dapat membuat pengecualian khusus untuk peran tersebut.

Anda juga dapat menerapkan kebijakan bucket ke semua atau bucket yang dipilih dengan menggunakan kebijakan seperti berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyNonHeaderAuth",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "StringNotEquals": {
          "s3:authType": "REST-HEADER",
          "aws:PrincipalTag/non-header-auth-allowed": "true"
        }
      }
    }
  ]
}
```

Kebijakan bucket ini memiliki efek menolak penggunaan CopyObjectdan UploadPartCopyAPI untuk membuat salinan lintas wilayah. Replikasi Amazon S3 tidak terpengaruh karena tidak bergantung pada API ini.

Jika Anda ingin menggunakan kebijakan bucket seperti kebijakan sebelumnya dan masih mendukung Cross-region CopyObjectatau UploadPartCopyAPI, tambahkan kondisi yang `aws:ViaAWSService` serupa dengan berikut ini:

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

```
  "Statement": [
    {
      "Sid": "DenyNonHeaderAuth",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::{bucket-name}/*",
      "Condition": {
        "StringNotEquals": {
          "s3:authType": "REST-HEADER",
          "aws:PrincipalTag/non-header-auth-allowed": "true"
        },
        "Bool": {
          "aws:ViaAWSService": "false"
        },
      }
    }
  ]
}
```

Menggabungkan pagar pembatas dan pengecualian untuk pagar pembatas lainnya

Jika Anda tidak berencana untuk menerapkan pagar pembatas secara umum kepada pengguna dan peran Anda, Anda mungkin ingin menerapkannya pada pengecualian pagar pembatas umum lainnya, sehingga pengecualian tersebut tidak mendukung permintaan yang telah ditetapkan sebelumnya.

Jika Anda memiliki batasan jaringan tetapi mengizinkan pengecualian untuk mitra eksternal atau kasus penggunaan khusus, Anda harus memblokir string kueri atau `POST` otentikasi saat pengecualian tersebut diterapkan, kecuali jika secara khusus diidentifikasi sebagai diperlukan.

Batasan untuk S3: SignatureAge

Administrator akan merasa berguna untuk memahami implikasi `s3:signatureAge` lebih lengkap. Setiap permintaan yang ditandatangani termasuk`X-Amz-Date`, yang harus menunjukkan waktu saat ini. Nilai ini diisi oleh klien dan penandatangan permintaan. AWS menolak permintaan yang dianggap memiliki waktu yang tidak valid. Namun, penandatangan dapat menghasilkan tanda tangan terlebih dahulu dengan waktu yang akan datang. Amazon S3 menolak permintaan yang menentukan masa depan jika dikirim terlalu jauh sebelumnya. Namun, jika permintaan tidak dikirim sampai waktu yang ditandatangani ke tanda tangan, tanda tangan dapat dibuat lebih awal dan dikirim kemudian.

`s3:signatureAge`membatasi usia maksimum `X-Amz-Date` dalam tanda tangan hanya untuk permintaan yang telah ditetapkan sebelumnya. Permintaan yang lebih tua dari usia yang ditentukan ditolak, bahkan jika kedaluwarsa `X-Amz-Expires` atau `POST` kebijakan akan menyatakannya valid. `s3:signatureAge`tidak mengubah periode valid untuk permintaan yang tidak menyertakan kedaluwarsa eksplisit. Itu juga tidak mengontrol nilai `X-Amz-Date` yang digunakan klien untuk tanda tangan.

Jika jam sistem salah atau jika klien sengaja melakukan permintaan tanggal di masa depan, waktu yang ditandatangani mungkin bukan waktu tanda tangan dibuat. Ini membatasi seberapa banyak yang `s3:signatureAge` dapat mengontrol solusi. Solusi yang menggunakan waktu saat ini ketika menghasilkan tanda tangan dibatasi dengan cara yang diharapkan: Tanda tangan tetap valid untuk jumlah milidetik yang ditentukan dalam. `s3:signatureAge` Solusi yang tidak menggunakan waktu saat ini akan memiliki batas yang berbeda. Salah satu batasan adalah bahwa kredensyal yang digunakan untuk menandatangani tanda tangan harus tetap valid. Sebagai administrator, Anda dapat mengontrol validitas maksimum kredensyal sementara yang dikeluarkan. Anda dapat mengizinkan kredensialnya valid hingga 36 jam atau membatasi validitas hingga serendah 15 menit. Kedaluwarsa kredensyal sementara tidak tergantung pada nilai. `X-Amz-Date`

Kredensyal permanen tidak memiliki batasan ini. Hanya menggunakan kredensyal sementara adalah praktik terbaik, dan Anda dapat secara eksplisit mencabut kredensi permanen apa pun, yang juga akan membatalkan tanda tangan apa pun berdasarkan kredensi tersebut.

Meskipun `s3:signatureAge` diukur dalam milidetik, tidak praktis untuk mengaturnya menjadi kurang dari 60 detik, bahkan jika Anda memiliki jam yang disinkronkan dengan baik dan penggunaan latensi rendah. Pengaturan yang lebih rendah dari 60 detik berisiko menolak permintaan yang valid. Jika Anda mengharapkan penundaan antara pembuatan tanda tangan dan pengiriman permintaan, atau masalah dengan sinkronisasi jam, Anda harus memperhitungkannya dalam pengelolaan. `s3:signatureAge`

Menargetkan ember dalam skala

SCP dan RCP dapat digunakan `aws:PrincipalTag` untuk membuat pengecualian bagi pengguna. Anda tidak dapat menggunakan tag pada bucket untuk mengontrol akses melalui `aws:ResourceTag` – hanya tag objek yang digunakan untuk kontrol akses. Umumnya tidak dapat diskalakan untuk menambahkan tag ke setiap objek yang ingin Anda terapkan kontrol ini.

Solusi yang sesuai dengan banyak kasus penggunaan adalah dengan menerapkan kebijakan dan pengecualian di tingkat akun, baik dengan mengubah akun yang diterapkan SCP atau RCP atau

dengan menggunakan aws:ResourceAccount, aws:ResourceOrgPaths, atau aws: ResourceOrg ID. Misalnya, SCP atau RCP dapat diterapkan ke satu set akun produksi.

Solusi lain adalah dengan menggunakan AWS Config aturan khusus untuk menerapkan kontrol detektif atau kontrol responsif. Tujuannya adalah agar setiap ember berisi kebijakan ember dengan pagar pembatas yang sesuai. Selain menguji konten kebijakan bucket, AWS Config aturan kustom dapat mengambil tag dari bucket dan mengecualikan bucket dari aturan jika bucket diberi tag dengan nilai tertentu. Jika aturan itu gagal dalam pemeriksaan kepatuhannya, aturan tersebut dapat menandai bucket sebagai tidak sesuai atau meminta perbaikan untuk menambahkan pagar pembatas ke kebijakan bucket.

> **Note**
>
> Anda tidak dapat membatasi konten tag permintaan. PutBucketTagging Untuk mempertahankan kontrol atas bagaimana bucket ditandai, Anda harus membatasi akses ke `PutBucketTagging` dan DeleteBucketTagging.

Interaksi dan mitigasi logging

URL presigned berisi tanda tangan dan dapat digunakan, selama periode sebelum kedaluwarsa, untuk melakukan operasi API tertentu yang ditandatangani. Ini harus diperlakukan sebagai kredensi akses sementara. Tanda tangan harus tetap pribadi hanya untuk pihak yang perlu mengetahuinya. Di sebagian besar lingkungan, ini adalah klien yang mengirim permintaan dan server yang menerimanya. Mengirim tanda tangan sebagai bagian dari sesi HTTPS langsung mempertahankan sifat pribadinya, karena hanya peserta sesi HTTPS yang memiliki visibilitas ke URI yang mentransmisikan tanda tangan.

Untuk URL yang telah ditetapkan sebelumnya, tanda tangan ditransmisikan sebagai `X-Amz-Signature` parameter string kueri. Parameter string kueri adalah komponen dari URI. Risikonya adalah klien dapat mencatat URI dan tanda tangan dengannya. Klien memiliki akses ke seluruh permintaan HTTP dan dapat mencatat bagian mana pun dari permintaan, data, dan header (termasuk header otentikasi). Namun, ini menurut konvensi kurang umum. Pencatatan URI lebih umum dan diperlukan dalam kasus seperti logging akses. Klien harus menggunakan redaksi atau masking untuk menghapus tanda tangan sebelum login URI.

Di beberapa lingkungan, pengguna mengizinkan perantara (proxy) untuk mendapatkan visibilitas ke sesi HTTPS mereka. Mengaktifkan proxy memerlukan tingkat akses istimewa yang tinggi ke

sistem klien, karena memerlukan konfigurasi dan sertifikat tepercaya. Instalasi konfigurasi proxy dan sertifikat tepercaya, dalam konteks lokal lingkungan perantara klien, memungkinkan tingkat hak istimewa yang sangat tinggi. Untuk alasan ini, akses ke perantara tersebut harus dikontrol dengan ketat.

Tujuan perantara biasanya untuk memblokir jalan keluar yang tidak diinginkan dan untuk melacak jalan keluar lainnya. Dengan demikian, biasanya perantara tersebut mencatat permintaan. Meskipun perantara dapat, seperti klien, mencatat konten, header, dan data apa pun (yang semuanya akan sangat sensitif), lebih umum bagi mereka untuk mencatat URI, seperti yang menyertakan parameter string `X-Amz-Signature` kueri.

Mitigasi

Kami menyarankan agar pencatatan URI menyunting parameter string `X-Amz-Signature` kueri, menyunting seluruh string kueri, atau memperlakukan informasi sebagai sangat rahasia, seperti dengan akses langsung ke server perantara. Meskipun perlindungan ini sangat direkomendasikan, fakta bahwa URL yang ditetapkan sebelumnya kedaluwarsa mengurangi risiko paparan log, selama eksposur tertunda cukup lama agar tanda tangan kedaluwarsa.

Amazon S3 juga melihat tanda tangan dan harus menanganinya dengan tepat. Log akses server Amazon S3 menyertakan URI permintaan tetapi menyuntingX-Amz-Signature, seperti yang disarankan. Hal yang sama berlaku ketika peristiwa CloudTrail data dicatat untuk Amazon S3. Anda dapat mengonfigurasi CloudWatch Log Amazon untuk menutupi data dengan menggunakan pengidentifikasi data khusus.

Ekspresi reguler berikut `X-Amz-Signature` cocok dengan yang muncul di URI:

```
X-Amz-Signature=[a-f0-9]{64}
```

Ekspresi reguler berikut ini menambahkan pola pengelompokan untuk mengidentifikasi teks yang akan diganti secara lebih spesifik:

```
(?:X-Amz-Signature=)([a-f0-9]{64})
```

Jika Anda memiliki entri log akses seperti berikut ini:

```
X-Amz-Signature=733255ef022bec3f2a8701cd61d4b371f3f28c9f193a1f02279211d48d5193d7
```

Ekspresi reguler pertama menerjemahkan entri log akses ke:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Ekspresi reguler kedua, pada sistem yang mendukung grup non-penangkapan, menerjemahkan entri log akses ke:

```
X-Amz-Signature=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Pertanyaan yang Sering Diajukan

Dapatkah permintaan yang telah ditetapkan sebelumnya digunakan beberapa kali? Apakah itu risiko keamanan?

Ya, tanda tangan dalam permintaan yang telah ditetapkan sebelumnya dapat digunakan lebih dari satu kali. Apakah ini risiko keamanan adalah pertanyaan kontekstual. Metode lain untuk mengakses layanan AWS memungkinkan pengulangan juga. Pengguna atau beban kerja dengan AWS kredensional dapat mengirim banyak permintaan ke Layanan AWS, dan salah satu permintaan tersebut dapat berupa duplikat.

Jika kasus penggunaan Anda memerlukan eksekusi sekali dan hanya sekali, Anda harus menerapkan mekanisme lain untuk menerapkan penggunaan tunggal. Penggunaan tunggal bukanlah fitur permintaan yang telah ditetapkan sebelumnya. Sebagai insinyur keamanan, Anda harus meninjau kasus penggunaan dan implementasi, tetapi dalam banyak kasus, beberapa penggunaan akan sesuai dengan penggunaan yang dapat diterima.

Dapatkah seseorang selain pengguna yang dituju menggunakan permintaan yang telah ditentukan sebelumnya?

Tanda tangan dalam permintaan yang telah ditentukan sebelumnya dapat dikirim oleh siapa saja yang memilikinya. Ini akan diterima hanya jika melewati bentuk validasi lain, seperti kontrol perimeter data. Jika tanda tangan telah kedaluwarsa, kredensi penandatanganan telah kedaluwarsa, atau kredensional penandatanganan tidak memiliki akses ke sumber daya yang diminta, permintaan akan ditolak.

Hal yang sama berlaku untuk metode otentikasi lainnya dengan Layanan AWS. Kredensi yang dibagikan secara tidak tepat memungkinkan akses yang tidak pantas. Praktik terbaik inti adalah berbagi kredensi dan tanda tangan hanya dengan audiens yang dituju. Jika Anda tidak dapat mempercayai audiens yang dituju untuk menjaga keamanan data pribadi dan tidak membagikannya dengan orang lain, ini akan merusak segala bentuk otentikasi.

Dapatkah pengguna yang berwenang menggunakan permintaan yang telah ditetapkan sebelumnya untuk mengekstrasi data?

Mengamankan data membutuhkan tindakan yang kuat. Mengaktifkan akses untuk tujuan yang dimaksudkan sambil mempertahankan perimeter data memerlukan pendekatan yang komprehensif. Akses dengan hak istimewa paling sedikit, kontrol perimeter data, dan hanya menggunakan kredenal akses sementara adalah praktik terbaik umum yang berlaku untuk mengamankan data. Penggunaan yang tepat dari kontrol ini juga membatasi kemampuan pengguna untuk melakukan tindakan melalui permintaan yang telah ditetapkan sebelumnya yang mereka hasilkan.

Hal ini karena akses yang disediakan oleh permintaan presigned adalah bagian dari akses yang diberikan ke kredensi yang digunakan untuk menandatangani permintaan. Dalam konteks ini, praktik terbaik yang berlaku untuk mengakses data umumnya berlaku untuk permintaan yang telah ditetapkan sebelumnya, tetapi permintaan yang telah ditetapkan sebelumnya tidak membuat akses baru ke data.

- Kedaluwarsa maksimum terbatas pada berakhirnya kredenal penandatanganan. Jika kredensi penandatanganan dicabut, tanda tangan berdasarkan kredensialnya tidak lagi valid.
- Jika izin untuk prinsipal IAM yang terkait dengan kredensi penandatanganan tidak menyertakan eksekusi tindakan yang terkait dengan permintaan yang telah ditetapkan sebelumnya, pemanggilan permintaan yang telah ditetapkan sebelumnya akan menghasilkan respons “akses ditolak”. Respons bergantung pada status izin saat ini pada saat pemanggilan, yang tidak memiliki hubungan dengan waktu tanda tangan permintaan yang telah ditetapkan sebelumnya dibuat.
- Properti prinsipal dievaluasi berdasarkan prinsip yang terkait dengan kredensi penandatanganan.
- Properti sesi peran dievaluasi berdasarkan sesi peran yang terkait dengan kredensi penandatanganan.
- Properti jaringan dievaluasi berdasarkan bagaimana permintaan diterima, seperti permintaan normal.

Dalam konteks ini, pemeriksaan risiko yang terkait dengan permintaan yang telah ditetapkan sebelumnya dibatasi pada area di mana mereka ditandatangani dengan kredensional yang berbeda dari kredensi pengguna dan menyediakan akses yang bukan bagian dari prinsipal pengguna. Pemeriksaan ini harus diterapkan pada desain layanan, beban kerja, atau solusi yang menghasilkan tanda tangan atas nama pengguna, bukan kemampuan permintaan yang telah ditentukan sebelumnya itu sendiri.

Dapatkah saya menolak akses dari URL yang telah ditetapkan sebelumnya jika saya curiga telah dibagikan dengan cara yang tidak sah?

Ya. Ini memerlukan pembatalan kredenal yang ditandatangani URL. Ada beberapa cara untuk mencapai ini:

- Hapus izin dari prinsipal IAM yang menjadi milik kredensialnya. Jika prinsipal IAM tersebut tidak lagi memiliki akses ke sumber daya dan operasi tempat URL ditandatangani, URL tidak dapat menjalankan operasi itu. Ini mempengaruhi semua penggunaan yang cocok dari prinsipal IAM tersebut.
- Jika kredensional yang digunakan untuk menandatangani URL adalah AWS STS kredensial sementara, Anda dapat mencabut izin sesi untuk kredenal sementara yang dikeluarkan sebelum waktu tertentu untuk prinsipal IAM. Bergantung pada kasus penggunaan, mungkin ada sesi valid lainnya yang menjadi tidak valid sebelum waktu kedaluwarsa normal, tetapi sesi baru tidak akan terpengaruh. Mencabut izin sesi juga membatalkan semua yang ditandatangani dengan menggunakan kredensional URLs yang terkait dengan sesi tersebut, tetapi yang baru terkait dengan sesi baru tidak akan URLs terpengaruh.
- Jika kredensional yang digunakan untuk menandatangani URL adalah kredensional permanen, nonaktifkan kunci akses. Ini memengaruhi semua penggunaan yang terkait dengan kredensi tersebut.

Sumber daya

Dokumentasi Amazon S3

- Permintaan Otentikasi (Versi AWS Tanda Tangan 4)
- Permintaan Otentikasi: Menggunakan Parameter Kueri (Versi AWS Tanda Tangan 4)
- Permintaan Otentikasi: Unggahan Berbasis Browser Menggunakan POST (AWS Versi Tanda Tangan 4)
- Amazon S3 Signature Versi 4 Kunci Kebijakan Khusus Otentikasi
- Bekerja dengan URL yang telah ditetapkan sebelumnya

Referensi lainnya

- Membangun Perimeter Data di AWS (AWS whitepaper)
- SEC03-BP02 Berikan akses hak istimewa paling sedikit (Kerangka yang Dirancang AWS dengan Baik, pilar Keamanan)
- SEC03-BP05 Tentukan pagar pembatas izin untuk organisasi Anda (Kerangka Kerja yang Dirancang dengan Baik, Pilar Keamanan)AWS

Lampiran A: Bagaimana penggunaan presigned Layanan AWS URLs

Lampiran ini memberikan informasi tentang Layanan AWS dan fitur yang menggunakan presigned. URLs Informasi ini melayani dua tujuan:

- Untuk menyediakan insinyur keamanan yang menerapkan kontrol dengan informasi tentang kemungkinan dampak dari kontrol tersebut.
- Untuk menciptakan kesadaran akan situasi di mana risiko ini mungkin relevan untuk interaksi pencatatan URL.

> **Important**
>
> Lampiran ini tidak menyediakan daftar lengkap Layanan AWS atau penggunaan presigned. URLs Ini juga tidak mencakup solusi kustom atau pihak ketiga.

Konsol Amazon S3

Prinsipal: Pengguna konsol

Kedaluwarsa default: 5 menit

> **Penafian**
>
> Bagian ini mendokumentasikan perilaku konsol Amazon S3 saat ini. AWS perilaku konsol dapat berubah tanpa pemberitahuan.

Konsol Amazon S3 mendukung pengunduhan dan pengunggahan objek. Unduhan menggunakan URL presigned yang memiliki waktu kedaluwarsa 300 detik (5 menit). URL dihasilkan oleh permintaan kehttps://<bucket-region>.console.aws.amazon.com/s3/batch0psServlet-proxy.

Permintaan tersebut dimulai ketika pengguna mengklik tombol unduh, sehingga URL tidak dibuat terlebih dahulu atau dikirim ke klien sampai permintaan eksplisit untuk mengunduh terjadi.

Unggahan serupa, kecuali bahwa konsol mengirimkan dua permintaan: `OPTIONS` sebagai pemeriksaan CORS pra-penerbangan, dan. `PUT` Kedua permintaan menggunakan tanda tangan yang sama.

Kredensil yang digunakan untuk penandatanganan adalah kredensil sementara yang terkait dengan pengguna yang saat ini masuk. Rincian tentang metode untuk mendapatkan kredensil sementara tersebut berada di luar cakupan panduan ini.

Amazon S3 Objek Lambda

Prinsipal: Penelepon titik akses

Kedaluwarsa default: 61 detik

> **Note**
>
> Per 7 November 2025, S3 Object Lambda hanya tersedia untuk pelanggan lama yang saat ini menggunakan layanan serta untuk memilih AWS Partner Network mitra (APN). Untuk kemampuan yang mirip dengan S3 Object Lambda, pelajari selengkapnya di sini — Perubahan ketersediaan Objek Amazon S3 Lambda.

Amazon S3 Object Lambda menggunakan AWS Lambda fungsi untuk memproses dan mengubah data secara otomatis saat diambil dari Amazon S3. Ketika S3 Object Lambda memanggil fungsi, fungsi tersebut disediakan URL `inputS3Url` () yang telah ditetapkan sebelumnya yang dapat digunakan untuk mengunduh objek asli dari titik akses pendukung.

Presigned ini ditandatangani URLs untuk jalur akses Amazon S3 pendukung, yang disediakan saat Anda mengonfigurasi S3 Object Lambda. (Ini tidak sama dengan titik akses Object Lambda.) Alih-alih menggunakan peran yang terikat pada fungsi Lambda, URL ditandatangani dengan menggunakan identitas pemanggil asli, dan izin pengguna tersebut akan berlaku saat URL digunakan. Jika ada header yang ditandatangani di URL, fungsi Lambda harus menyertakan header ini dalam panggilan ke Amazon S3.

URL presigned yang dikembalikan memiliki waktu kedaluwarsa 61 detik (satu detik lebih dari durasi maksimum untuk fungsi Lambda Objek S3). URL yang dihasilkan hanya dapat digunakan dengan titik akses pendukung. Penelepon titik akses Lambda Objek S3 perlu memiliki akses ke titik akses ini. Anda dapat membatasi akses itu ke konteks S3 Object Lambda dengan menggunakan kondisi. `"aws:CalledVia": ["s3-object-lambda.amazonaws.com"]` Ketika kondisi tersebut

dilampirkan ke titik akses atau bucket pendukung, pengguna tidak dapat mengakses jalur akses atau bucket pendukung secara langsung.

Nilai dari pendekatan ini adalah tidak perlu memberikan akses fungsi Lambda ke bucket atau titik akses S3 Anda. Peran yang terkait dengan fungsi Lambda akan memerlukan izin untuk WriteGetObjectResponse, tetapi tidak memerlukan izin untuk. GetObject

Ketika S3 Object Lambda menghasilkan URLs presigned, itu tidak menambahkan batasan jaringan, sehingga URL dapat digunakan di luar fungsi Lambda. Namun, batasan apa pun yang ditempatkan pada penelepon S3 Object Lambda masih berlaku. Misalnya, jika fungsi Lambda Anda berjalan di VPC dan Anda membatasi pemanggil untuk menggunakan titik akhir VPC, siapa pun yang memiliki URL yang telah ditentukan sebelumnya akan memerlukan kemampuan untuk mengirimkannya melalui titik akhir VPC tersebut. Pembatasan ini juga berlaku untuk SourceIpdan VpcSourceIp.

> **Note**
>
> Untuk menggunakan fungsi Lambda Objek S3 di VPC, VPC harus memiliki rute ke titik akhir S3 publik untuk dipanggil. WriteGetObjectResponse Ini tidak menunjukkan bahwa persyaratan untuk menggunakan titik akhir VPC tidak akan berlaku untuk permintaan untuk mengambil data dari bucket.

AWS Lambda Lintas Wilayah CopyObject

Prinsipal:AWS internal

Kedaluwarsa default: 3600 detik

Saat Anda menggunakan CopyObjectatau UploadPartCopyAPI untuk menyalin Wilayah AWS, Amazon S3 menggunakan URLs presigned internal. Ini APIs dapat dipanggil langsung dari SDKs atau dari AWS CLI perintah `aws s3api copy-object` dan`aws s3api upload-part`. Ini APIs tidak digunakan untuk Replikasi Amazon S3, tetapi digunakan oleh `aws s3 sync` perintah AWS CLI `aws s3 cp` dan saat sumber dan tujuan adalah bucket S3. Mereka juga didukung oleh `TransferManager` implementasi di berbagai AWS SDKs.

AWS Lambda GetFunction

Prinsipal: AWS internal

Kedaluwarsa default: 10 menit

AWS Lambda menyimpan versi pengguna dalam bucket S3 yang dimiliki tim Lambda, sebelum membuat aset yang digunakan ke kontainer Lambda. Ketika Anda ingin mengakses kode untuk fungsi Anda, Anda memanggil GetFunctionAPI. API ini merespons dengan`Code.Location`, yang berisi URL yang telah ditetapkan sebelumnya yang valid selama 10 menit (waktu kedaluwarsa ini adalah perilaku saat ini dan bukan kontrak yang dipublikasikan). Jika Anda tidak ingin kode, Anda dapat menggunakan kombinasi GetFunctionConfiguration, GetFunctionConcurrency, dan ListTagsuntuk mengambil data lain yang dikembalikan oleh`GetFunction`.

URL yang dikembalikan tidak ditandatangani dengan kredensil pengguna yang saat ini masuk, tetapi atas nama pengguna oleh Lambda. Untuk alasan ini, kunci kondisi (seperti`aws:SourceIP`) yang diterapkan ke pengguna yang saat ini masuk atau kredensi sesi sementara pengguna tidak berlaku untuk URL yang dihasilkan. Ini benar apakah kunci kondisi GetFunctionhanya diterapkan, atau diterapkan ke semua penggunaan AWS API untuk pengguna atau sesi.

Konsol Lambda juga menggunakan GetFunctiondan URL presigned yang dikembalikan. Konsol menggunakan kredensil sementara yang terkait dengan pengguna yang saat ini masuk untuk menelepon. GetFunction Rincian tentang mendapatkan kredensil sementara tersebut berada di luar cakupan dokumen ini.

Amazon ECR

Prinsipal: AWS internal

Kedaluwarsa default: 1 jam

Amazon Elastic Container Registry (Amazon ECR) menyediakan GetDownloadUrlForLayerAPI, yang mengembalikan URL yang telah ditetapkan sebelumnya yang valid selama satu jam dan mendukung pengunduhan satu lapisan dari gambar Amazon ECR. Namun, operasi ini digunakan oleh proxy Amazon ECR dan umumnya tidak digunakan oleh pengguna untuk menarik dan mendorong gambar.

Amazon Redshift Spectrum

Prinsip: Peran diteruskan ke CREATE EXTERNAL SCHEMA melalui `IAM_ROLE`

Kedaluwarsa default: 1 jam

Amazon Redshift Spectrum menggunakan URLs presigned internal dan melarang pembatasan kombinasi bucket dan peran Amazon Redshift yang akan membatasi presigned. URLs Anda dapat menggunakan `s3:signatureAge` nilai 16 menit, tetapi nilai yang sangat rendah tidak dapat diandalkan. Nilai minimum yang dapat Anda gunakan tergantung pada waktu dan ukuran kueri Anda. Meskipun nilai yang lebih rendah dari 16 menit berfungsi untuk banyak skenario, itu membutuhkan pengujian. Peran tersebut dapat dan harus dibatasi untuk digunakan hanya oleh Redshift Spectrum, yang tidak mengungkapkan yang URLs dihasilkannya, sehingga mengurangi pembenaran khas untuk nilai kedaluwarsa yang lebih rendah.

Studio SageMaker AI Amazon

Amazon SageMaker AI Studio mendukung dua tindakan API: CreatePresignedDomainUrldan CreatePresignedNotebookInstanceUrl. Namun, ini APIs tidak terkait dengan fitur URL presigned Signature Version 4. Ini APIs membuat URL yang menggunakan `authToken` parameter, tetapi tidak mendukung parameter kueri Signature Version 4 standar apa pun.

`authToken`adalah mekanisme yang berbeda tetapi memiliki kesamaan dengan URLs presigned. Ini dikirim sebagai parameter string kueri dan mendukung waktu kedaluwarsa 5 menit.

SageMaker AI mendukung pembatasan jaringan. Jika Anda membatasi `sagemaker:CreatePresignedDomainUrl` tindakan, tindakan tersebut berlaku baik untuk panggilan CreatePresignedDomainUrlmaupun penggunaan URL yang dihasilkan. Jika URL dihasilkan dari jaringan yang valid dan kemudian dikirim oleh jaringan yang tidak valid, panggilan API untuk menghasilkan URL berhasil, tetapi permintaan yang mengirim URL gagal. Hal yang sama berlaku CreatePresignedNotebookInstanceUrldan `sagemaker:CreatePresignedNotebookInstanceUrl` tindakannya.

Untuk informasi selengkapnya, lihat dokumentasi SageMaker AI.

Lampiran B: Bagaimana kontrol untuk pengaruh yang telah ditentukan sebelumnya URLs Layanan AWS

Lampiran ini menjelaskan interaksi antara Layanan AWS yang menggunakan presigned URLs, seperti yang dijelaskan dalam Lampiran A, dan kontrol yang dijelaskan sebelumnya dalam panduan ini.

Pagar pembatas untuk S3: SignatureAge

Konsol Amazon S3 tidak terganggu oleh kedaluwarsa maksimum 5 menit yang ditetapkan oleh tombol kondisi. `s3:signatureAge` Konsol Amazon S3 menghasilkan presigned URLs ketika Anda memilih tombol Unduh dan menerapkan waktu kedaluwarsa 5 menitnya sendiri. Durasi maksimum yang lebih pendek dari 2 menit dapat membuat kegagalan acak berdasarkan sinkronisasi jam dan latensi.

Amazon S3 Object Lambda menggunakan waktu kedaluwarsa 61 detik, jadi pengaturan kondisi pada `s3:signatureAge` nilai 61 detik atau lebih tidak akan menyebabkan gangguan apa pun. Durasi yang lebih pendek mungkin kurang dapat diandalkan dan dapat menyebabkan kegagalan intermiten.

Amazon S3 Cross-region CopyObjecttidak terganggu oleh kedaluwarsa maksimum 5 menit. Namun, durasi yang lebih pendek dapat membuat kegagalan acak berdasarkan sinkronisasi jam dan latensi.

Di AWS Lambda, GetFunctionberikan URL ke objek di luar akun pelanggan, sehingga kebijakan pelanggan tidak memengaruhi yang dihasilkan URLs.

Amazon Redshift Spectrum telah diuji `s3:signatureAge` dengan kondisi 16 menit. Namun, durasi yang lebih pendek dapat menyebabkan gangguan.

Guardrail untuk S3:authType saat tidak menggunakan batasan jaringan

Konsol Amazon S3 biasanya dipengaruhi oleh pagar `s3:authType` pembatas. Konsol merutekan ke Amazon S3 berdasarkan konfigurasi jaringan lokal. Jika jaringan lokal merutekan ke Amazon S3 dengan cara yang memungkinkan pembatasan jaringan, konsol Amazon S3 akan tetap berfungsi. Namun, jika dialihkan melalui proxy atau internet publik dengan cara yang tidak diizinkan,

penggunaan akan diblokir. Namun, memblokir penggunaan mungkin merupakan maksud dari kebijakan ini.

Lambda Objek Amazon S3 terpengaruh jika fungsi Lambda tidak terhubung ke VPC yang sesuai. Dalam konfigurasi ini, VPC harus memiliki gateway NAT, bukan untuk mengakses bucket S3, tetapi untuk menelepon. WriteGetObjectResponse

Amazon S3 Lintas wilayah CopyObjectterganggu jika pagar pembatas ini diterapkan pada kebijakan bucket tanpa pengecualian yang disarankan untuk kapan benar. **`aws:viaAWSService`**

Amazon Redshift Spectrum dipengaruhi oleh pagar pembatas kecuali `s3:authType` perutean VPC yang ditingkatkan digunakan. Saat ini, Redshift Spectrum mendukung peningkatan perutean VPC hanya dengan cluster tanpa server, bukan dengan cluster yang disediakan.

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Pembaruan dan klarifikasi	Di bagian pagar pembatas tambahan, menambahkan informasi tentang RCPs dan mengklarifikasi pengecualian.	Agustus 8, 2025
Publikasi awal	—	Juli 23, 2024

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