Benutzerhandbuch

AWS Verifizierter Zugriff

AWS Verifizierter Zugriff: Benutzerhandbuch

Table of Contents

Was ist AWS verifizierter Zugriff?

Mit AWS Verified Access können Sie sicheren Zugriff auf Ihre Anwendungen gewähren, ohne ein virtuelles privates Netzwerk (VPN) verwenden zu müssen. Verified Access bewertet jede Anwendungsanfrage und stellt sicher, dass Benutzer nur dann auf jede Anwendung zugreifen können, wenn sie die angegebenen Sicherheitsanforderungen erfüllen.

Vorteile von Verified Access

- Verbesserter Sicherheitsstatus — Ein herkömmliches Sicherheitsmodell bewertet den Zugriff einmal und gewährt dem Benutzer Zugriff auf alle Anwendungen. Verified Access wertet jede Anwendungszugriffsanfrage in Echtzeit aus. Dies macht es schlechten Akteuren schwer, von einer Anwendung zur anderen zu wechseln.
- Integration mit Sicherheitsdiensten — Verified Access lässt sich in Identitäts- und Geräteverwaltungsdienste integrieren, einschließlich Dienste von Drittanbietern. AWS Anhand von Daten aus diesen Diensten überprüft Verified Access die Vertrauenswürdigkeit von Benutzern und Geräten anhand einer Reihe von Sicherheitsanforderungen und bestimmt, ob der Benutzer Zugriff auf eine Anwendung haben sollte.
- Verbesserte Benutzererfahrung — Mit Verified Access müssen Benutzer kein VPN mehr verwenden, um auf Ihre Anwendungen zuzugreifen. Dies trägt dazu bei, die Anzahl der Supportfälle zu reduzieren, die sich aus VPN-Problemen ergeben.
- Vereinfachte Problembehebung und Audits — Verified Access protokolliert alle Zugriffsversuche und bietet so einen zentralen Überblick über den Anwendungszugriff, sodass Sie schnell auf Sicherheitsvorfälle und Prüfungsanfragen reagieren können.

Zugriff auf AWS verifizierten Zugriff

Sie können jede der folgenden Schnittstellen verwenden, um mit Verified Access zu arbeiten:

- AWS Management Console— Bietet eine Webschnittstelle für die Erstellung und Verwaltung von Verified Access-Ressourcen. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
- AWS Command Line Interface(AWS CLI) — Stellt Befehle für eine Vielzahl von Befehlen bereitAWS-Services, einschließlich AWS Verified Access. Die AWS CLI wird unter Windows,

macOS und Linux unterstützt. Informationen zum Aufrufen der AWS CLI finden Sie unter AWS Command Line Interface.

- AWSSDKs — Geben sprachspezifische APIs an. Die AWS SDKs kümmern sich um viele der Verbindungsdetails, wie z. B. die Berechnung von Signaturen und die Verarbeitung von Anforderungswiederholungen und Fehlern. Weitere Informationen finden Sie unter AWS-SDKs.
- Abfrage-API – Bietet API-Aktionen auf niedriger Ebene, die Sie mithilfe von HTTPS-Anforderungen aufrufen. Die Verwendung der Abfrage-API ist die direkteste Möglichkeit für den Zugriff auf Verified Access. Allerdings muss Ihre Anwendung die Details auf niedriger Ebene behandeln, z. B. die Generierung des Hashs zum Signieren der Anforderung und die Fehlerbehandlung. Weitere Informationen finden Sie unter Verified Access-Aktionen in der Amazon EC2 EC2-API-Referenz.

Dieser Leitfaden beschreibt, wie Sie die verwenden, AWS Management Console um die Ressourcen für die Erstellung von Verified Access zu erstellen, den Zugriff darauf und deren Verwaltung.

Preisgestaltung

Jede Anfügung an ein verarbeitete Datenverkehr wird Ihnen stündlich berechnet, und Ihnen wird der von Verified Access verarbeitete Datenverkehr in Rechnung gestellt. Weitere Informationen finden Sie unter Preise für den AWS verifizierten Zugriff auf.

So funktioniert Verified Access

AWSVerified Access wertet jede Anwendungsanfrage Ihrer Benutzer aus und ermöglicht den Zugriff auf der Grundlage von:

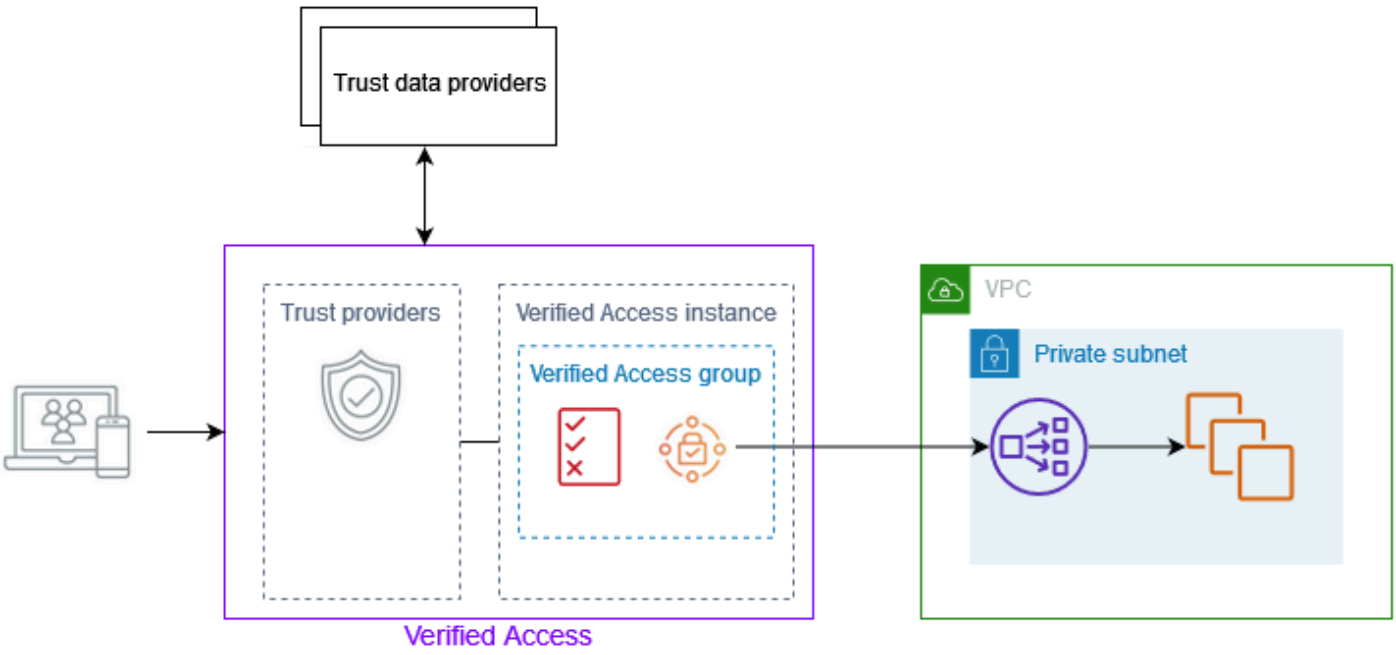
- Vertrauensdaten, die von Ihrem ausgewählten Vertrauensanbieter (von AWS oder einem Dritten) gesendet wurden.
- Greifen Sie auf Richtlinien zu, die Sie in Verified Access erstellen.

Wenn ein Benutzer versucht, auf eine Anwendung zuzugreifen, erhält Verified Access seine Daten vom Vertrauensanbieter und wertet sie anhand der Richtlinien aus, die Sie für die Anwendung festgelegt haben. Verified Access gewährt nur dann Zugriff auf die angeforderte Anwendung, wenn der Benutzer Ihre angegebenen Sicherheitsanforderungen erfüllt. Alle Anwendungsanfragen werden standardmäßig abgelehnt, bis eine Richtlinie definiert ist.

Darüber hinaus protokolliert Verified Access jeden Zugriffsversuch, sodass Sie schnell auf Sicherheitsvorfälle und Prüfungsanfragen reagieren können.

Die wichtigsten Komponenten von Verified Access

Das folgende Diagramm bietet einen allgemeinen Überblick über Verified Access. Benutzer senden Anfragen, um auf eine Anwendung zuzugreifen. Verified Access bewertet die Anfrage anhand der Zugriffsrichtlinie für die Gruppe und aller anwendungsspezifischen Endpunktrichtlinien. Wenn der Zugriff erlaubt ist, wird die Anfrage über den Endpunkt an die Anwendung gesendet.



- Instanzen mit verifiziertem Zugriff — Eine Instanz wertet Anwendungsanfragen aus und gewährt Zugriff nur, wenn Ihre Sicherheitsanforderungen erfüllt sind.
- Verifizierte Zugriffsendpunkte — Jeder Endpunkt steht für eine Anwendung. Sie können einen Load Balancer-Endpunkt oder einen Netzwerkschnittstellenendpunkt erstellen.
- Gruppe „Verifizierter Zugriff“ — Eine Sammlung von Verified Access-Endpunkten. Wir empfehlen, die Endpunkte für Anwendungen mit ähnlichen Sicherheitsanforderungen zu gruppieren, um die Richtlinienverwaltung zu vereinfachen. Sie können beispielsweise die Endpunkte für all Ihre Vertriebsanwendungen zusammenfassen.
- Zugriffsrichtlinien — Eine Reihe benutzerdefinierter Regeln, die festlegen, ob der Zugriff auf eine Anwendung zugelassen oder verweigert werden soll. Sie können eine Kombination von Faktoren angeben, darunter die Benutzeridentität und den Sicherheitsstatus des Geräts. Sie erstellen für jede Gruppe mit verifiziertem Zugriff eine Gruppenzugriffsrichtlinie, die von allen Endpunkten in der Gruppe übernommen wird. Sie können optional anwendungsspezifische Richtlinien erstellen und diese an bestimmte Endpunkte anhängen.
- Trust Providers — Ein Dienst, der Benutzeridentitäten oder den Sicherheitsstatus des Geräts verwaltet. Verified Access arbeitet AWS sowohl mit Vertrauensanbietern als auch mit Drittanbietern zusammen. Sie müssen jeder Verified Access-Instanz mindestens einen Vertrauensanbieter zuordnen. Sie können jeder Verified Access-Instanz einen einzelnen Identity Trust Provider und mehrere Device Trust Provider zuordnen.

- Vertrauensdaten — Die sicherheitsrelevanten Daten für Benutzer oder Geräte, die Ihr Vertrauensanbieter an Verified Access sendet. Wird auch als Benutzeransprüche oder Vertrauenskontext bezeichnet. Zum Beispiel die E-Mail-Adresse eines Benutzers oder die Betriebssystemversion eines Geräts. Verified Access wertet diese Daten anhand Ihrer Zugriffsrichtlinien aus, wenn es jede Anfrage zum Zugriff auf eine Anwendung erhält.

Tutorial: Erste Schritte mit Verified Access

Verwenden Sie dieses Tutorial, um mit AWS Verified Access zu beginnen. Sie erfahren, wie Sie Ressourcen mit verifiziertem Zugriff erstellen und konfigurieren.

Bevor Sie diese Anwendung zu Verified Access hinzufügen, war die Anwendung nur über Ihr privates Netzwerk zugänglich. Am Ende dieses Tutorials können bestimmte Benutzer über das Internet auf dieselbe Anwendung zugreifen, ohne VPN zu verwenden.

> **Note**
>
> Dieses Beispiel zeigt keine Integration mit Ihrem gerätebasierten Vertrauensanbieter. In diesem Beispiel arbeiten wir nur mit einem identitätsbasierten Vertrauensanbieter.

Aufgaben

Voraussetzungen

Die folgenden Voraussetzungen gelten für dieses Tutorial:

- Um dieses Beispiel für die Verwendung von Verified Access zu demonstrieren, verwenden wir zwei AWS-Konten. Ein Konto hostet Ihre Zielanwendung und die Ressourcen für verifizierten Zugriff werden im anderen Konto erstellt.
- Aktivieren Sie AWS IAM Identity Center in der AWS-Region, in der Sie arbeiten. Anschließend können Sie IAM Identity Center als Vertrauensanbieter mit Verified Access verwenden. Weitere Informationen finden Sie unter IAM Identity Center aktivieren im AWS IAM Identity Center - Benutzerhandbuch.
- Eine öffentlich gehostete Domain und die erforderlichen Berechtigungen zum Aktualisieren von DNS-Datensätzen für die Domain.
- Eine Anwendung, die hinter einem internen Load Balancer in einer ausgeführt wirdAWS-Konto. Der Anwendungsdomänenname, den wir verwenden werden, ist `www.myapp.example.com`.
- Stellen Sie sicher, dass Ihre IAM-Richtlinie über alle erforderlichen Berechtigungen verfügt, um eine hier aufgeführte AWS Instance mit verifiziertem Zugriff zu erstellenRichtlinie für die Erstellung von Instanzen mit verifiziertem Zugriff.

Schritt 1: Erstellen einer Instance mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um eine Instance mit verifiziertem Zugriff zu erstellen.

So erstellen Sie eine Instance mit verifiziertem Zugriff

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Amazon-VPC-Navigationsbereich Instances mit verifiziertem Zugriff und dann Instance mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie unter Name und Beschreibung einen Namen und eine Beschreibung für die Instance mit verifiziertem Zugriff ein.
4. Behalten Sie für Vertrauensanbieter die Standardoption bei.
5. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
6. Wählen Sie Verified Access Instance erstellen aus.

Schritt 2: Konfigurieren eines Vertrauensanbieters

Sie können AWS IAM Identity Center als Ihren Vertrauensanbieter einrichten.

So erstellen Sie einen IAM-Identity-Center-Vertrauensanbieter

1. Wählen Sie im Amazon-VPC-Navigationsbereich Vertrauensanbieter für verifizierten Zugriff und dann Vertrauensanbieter für verifizierten Zugriff erstellen aus.
2. (Optional) Geben Sie unter Namens-Tag und Beschreibung einen Namen und eine Beschreibung für den Verified Access Trust Provider ein.
3. Geben Sie eine benutzerdefinierte Kennung ein, die später bei der Arbeit mit Richtlinienregeln für den Richtlinienreferenznamen verwendet werden soll. Sie können beispielsweise eingeben**`idc`**.
4. Wählen Sie unter Vertrauensanbietertyp die Option Benutzervertrauensanbieter aus.
5. Wählen Sie unter Benutzervertrauensanbietertyp die Option IAM Identity Center aus.
6. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
7. Wählen Sie Vertrauensanbieter für verifizierten Zugriff erstellen aus.

Schritt 3: Anfügen Ihres Vertrauensanbieters an die Instance

Gehen Sie wie folgt vor, um den Vertrauensanbieter an Ihre Instance mit verifiziertem Zugriff anzuhängen.

So fügen Sie Ihrer Instance einen Vertrauensanbieter an

1. Wählen Sie im Amazon-VPC-Navigationsbereich Instances mit verifiziertem Zugriff aus.
2. Wählen Sie Ihre Instance aus.
3. Wählen Sie Aktionen, Vertrauensanbieter für verifizierten Zugriff anfügen aus.
4. Wählen Sie für Vertrauensanbieter für verifizierten Zugriff Ihren Vertrauensanbieter aus.
5. Wählen Sie Vertrauensanbieter für verifizierten Zugriff anfügen aus.

Schritt 4: Erstellen einer Gruppe mit verifiziertem Zugriff

Erstellen wir eine Gruppe, die Sie für den Endpunkt verwenden können, den Sie im nächsten Schritt erstellen werden.

So erstellen Sie eine Gruppe mit verifiziertem Zugriff

1. Wählen Sie im Amazon-VPC-Navigationsbereich Verified Access Groups und dann Create Verified Access Group aus.
2. (Optional) Geben Sie unter Namens-Tag und Beschreibung einen Namen und eine Beschreibung für die Gruppe ein.
3. Wählen Sie für Instance mit verifiziertem Zugriff Ihre Instance mit verifiziertem Zugriff aus.
4. Lassen Sie für Richtliniendefinition dieses Feld leer. Sie erstellen später in diesem Tutorial eine Richtlinie.
5. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
6. Wählen Sie Verified Access Group erstellen aus.

Schritt 5: Ihre Gruppe mit verifiziertem Zugriff über freigeben AWS Resource Access Manager

In diesem Schritt geben Sie die soeben erstellte Gruppe für die frei, AWS-Konto in der Ihre Zielanwendung ausgeführt wird. Um eine Gruppe mit verifiziertem Zugriff freizugeben, müssen Sie sie einer Ressourcenfreigabe hinzufügen. Wenn Sie keine Ressourcenfreigabe haben, müssen Sie zuerst eine erstellen.

Wenn Sie Teil einer Organisation in sind AWS Organizationsund die Freigabe innerhalb Ihrer Organisation aktiviert ist, wird Konsumenten in Ihrer Organisation automatisch Zugriff auf die freigegebene Gruppe Verified Access gewährt. Andernfalls erhalten Konsumenten eine Einladung zur Teilnahme an der Ressourcenfreigabe und nach Annahme der Einladung wird ihnen Zugriff auf die freigegebene Gruppe Verified Access gewährt.

Führen Sie die Schritte unter Erstellen einer Ressourcenfreigabe im AWS RAM-Benutzerhandbuch aus. Wählen Sie für Ressourcentyp auswählen die Option Verifizierter Zugriffsgruppe aus und aktivieren Sie dann das Kontrollkästchen für Ihre Gruppe mit verifiziertem Zugriff.

Weitere Informationen finden Sie unter Erste Schritte im AWS RAM-Benutzerhandbuch.

Schritt 6: Hinzufügen Ihrer Anwendung durch Erstellen eines Endpunkts

Gehen Sie wie folgt vor, um einen Endpunkt zu erstellen. In diesem Schritt wird davon ausgegangen, dass eine Anwendung hinter einem internen Load Balancer von Elastic Load Balancing ausgeführt wird.

So erstellen Sie einen Endpunkt mit verifiziertem Zugriff

1. Wählen Sie im Amazon-VPC-Navigationsbereich Verified Access-Endpunkte und dann Verified Access-Endpunkt erstellen aus.
2. (Optional) Geben Sie unter Namens-Tag und Beschreibung einen Namen und eine Beschreibung für den Endpunkt ein.
3. Wählen Sie für Verified Access Group Ihre Verified Access Group aus.
4. Gehen Sie für Anwendungsdetails wie folgt vor:
    a. Geben Sie für Anwendungsdomäne einen DNS-Namen für Ihre Anwendung ein.
    b. Wählen Sie unter Domänenzertifikat-ARN den Amazon-Ressourcennamen (ARN) Ihres öffentlichen TLS-Zertifikats aus.
5. Gehen Sie für Endpunktdetails wie folgt vor:
    a. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.
    b. Wählen Sie für Sicherheitsgruppen eine Sicherheitsgruppe aus, die dem Endpunkt zugeordnet werden soll.
    c. Geben Sie für Endpunkt-Domainpräfix eine benutzerdefinierte Kennung ein. Dies wird dem DNS-Namen vorangestellt, den Verified Access generiert. In diesem Beispiel können wir verwenden**`my-ava-app`**.
    d. Wählen Sie für Endpunkttyp die Option Load Balancer aus.
    e. Wählen Sie für Protokoll die Option HTTPS oder HTTP aus. Dies hängt von der Konfiguration Ihres Load Balancers ab.
    f. Geben Sie unter Port die Portnummer ein. Dies hängt von der Konfiguration Ihres Load Balancers ab.
    g. Wählen Sie für Load Balancer-ARN Ihren Load Balancer aus.
    h. Wählen Sie für Subnetze die Subnetze aus, die Ihrem Load Balancer zugeordnet sind.

6. Geben Sie für Richtliniendefinition derzeit keine Richtlinie ein. Wir werden dies später im Tutorial behandeln.
7. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
8. Wählen Sie Verified-Access-Endpunkt erstellen aus.

Schritt 7: Konfigurieren von DNS-Einstellungen

In diesem Schritt ordnen Sie den Domänennamen Ihrer Anwendung (z. B. www.myapp.example.com) dem Domänennamen Ihres Endpunkts mit verifiziertem Zugriff zu. Um die DNS-Zuweisung abzuschließen, erstellen Sie einen Canonical Name Record (CNAME) bei Ihrem DNS-Anbieter. Nachdem Sie den CNAME-Datensatz erstellt haben, werden alle Anfragen von Benutzern an Ihre Anwendung an Verified Access gesendet.

So rufen Sie den Domänennamen Ihres Endpunkts ab

1. Wählen Sie im Amazon-VPC-Navigationsbereich Verified Access-Endpunkte aus.
2. Wählen Sie den Endpunkt aus, den Sie zuvor erstellt haben.
3. Wählen Sie die Registerkarte Details für den Endpunkt aus.
4. Kopieren Sie die Endpunktdomäne aus unter Endpunktdomäne .

In diesem Tutorial lautet der Domänenname des Endpunkts `my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`.

Erstellen Sie einen CNAME-Datensatz bei Ihrem DNS-Anbieter:

Datensatzname	Typ	Wert
www.myapp.example.com	CNAME	my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com

Schritt 8: Testen der Konnektivität zu Ihrer Anwendung

Sie können jetzt die Konnektivität zu Ihrer Anwendung testen. Geben Sie den Domänennamen Ihrer Anwendung in Ihren Webbrowser ein. Das Standardverhalten von Richtlinien für verifizierten Zugriff besteht darin, alle Anfragen abzulehnen. Da wir noch keine Richtlinie eingerichtet haben, die jedem Zugriff erlauben würde, sollten alle Anfragen abgelehnt werden.

Schritt 9: Konfigurieren der Zugriffsrichtlinie auf Gruppenebene

Gehen Sie wie folgt vor, um die Gruppe Verified Access zu ändern und eine Zugriffsrichtlinie zu konfigurieren, die Konnektivität zu Ihrer Anwendung zulässt. Die Details der Richtlinie hängen von den Benutzern und Gruppen ab, die im IAM Identity Center konfiguriert sind. Informationen zum Erstellen einer Richtlinie finden Sie unter Verifizierte Zugriffsrichtlinien.

So ändern Sie eine Gruppe mit verifiziertem Zugriff

1. Wählen Sie im Amazon-VPC-Navigationsbereich Verified Access Groups aus.
2. Wählen Sie die -Gruppe aus.
3. Wählen Sie Aktionen, Gruppenrichtlinie für verifizierten Zugriff ändern aus.
4. Geben Sie die Richtlinie ein.
5. Wählen Sie Gruppenrichtlinie für verifizierten Zugriff ändern aus.

Schritt 10: Konnektivität erneut testen

Nachdem Ihre Gruppenrichtlinie eingerichtet ist, können Sie auf Ihre Anwendung zugreifen. Geben Sie den Domänennamen Ihrer Anwendung in Ihren Webbrowser ein. Die Anforderung sollte zulässig sein und Sie sollten zur Anwendung umgeleitet werden.

Bereinigen

Nachdem Sie mit dem Testen fertig sind, führen Sie den folgenden Schritt aus, um die erstellten Ressourcen zu löschen.

So löschen Sie die mit diesem Tutorial erstellten Ressourcen für verifizierten Zugriff

1. Wählen Sie im Amazon-VPC-Navigationsbereich Verified Access-Endpunkte aus. Wählen Sie den Endpunkt aus, den Sie entfernen möchten. Wählen Sie Aktionen, Verifizierten Zugriff-Endpunkt löschen aus.
2. Wählen Sie im Navigationsbereich Verified Access Groups aus. Wählen Sie die Gruppe aus, die Sie entfernen möchten. Wählen Sie Aktionen, Verifizierte Zugriffsgruppe löschen aus. Hinweis: Möglicherweise müssen Sie einige Minuten warten, bis der Vorgang zum Löschen des Endpunkts abgeschlossen ist.
3. Wählen Sie im Amazon-VPC-Navigationsbereich Instances mit verifiziertem Zugriff aus. Wählen Sie die Instance aus, die Sie für dieses Tutorial erstellt haben. Wählen Sie Aktionen, Vertrauensanbieter für verifizierten Zugriff trennen aus. Wählen Sie den Vertrauensanbieter aus der Dropdown-Liste aus und wählen Sie Vertrauensanbieter für verifizierten Zugriff trennen aus.
4. Wählen Sie im Navigationsbereich von Amazon VPC die Option Vertrauensanbieter für verifizierten Zugriff aus. Wählen Sie den Vertrauensanbieter aus, den Sie für dieses Tutorial erstellt haben. Wählen Sie Aktionen, Vertrauensanbieter für verifizierten Zugriff löschen aus.
5. Wählen Sie im Amazon-VPC-Navigationsbereich Verified Access Instances aus. Wählen Sie die Instance aus, die Sie für dieses Tutorial erstellt haben. Wählen Sie Aktionen, Instance mit verifiziertem Zugriff löschen aus.

Verifizierte Access-Instanzen

Eine AWS Verified Access-Instanz ist eine AWS Ressource, die Ihnen hilft, Ihre Vertrauensanbieter und Verified Access-Gruppen zu organisieren.

Themen

Erstellen Sie eine Instanz mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um eine Verified Access-Instanz zu erstellen.

Um eine Verified Access-Instanz zu erstellen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances und dann Create Verified Access-Instanz aus.
3. (Optional) Geben Sie unter Name und Beschreibung einen Namen und eine Beschreibung für die Verified Access-Instanz ein.
4. (Optional) Wählen Sie „Aktivieren“ für Federal Information Process Standards (FIPS), wenn Verified Access FIPS-konform sein muss.
5. (Optional) Wählen Sie für Trust Provider einen Trust Provider aus, der an die Verified Access-Instanz angehängt werden soll.
6. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
7. Wählen Sie Create Verified Access-Instanz aus.

Ordnen Sie einer Instanz einen Vertrauensanbieter zu

Gehen Sie wie folgt vor, um einen Trust Provider an eine Verified Access-Instanz anzuhängen.

So fügen Sie einer Verified Access-Instanz einen Vertrauensanbieter hinzu

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances aus.
3. Wählen Sie die Instance aus.
4. Wählen Sie Aktionen, Vertrauensanbieter mit verifiziertem Zugriff anhängen aus.
5. Wählen Sie unter Vertrauensanbieter mit verifiziertem Zugriff einen Vertrauensanbieter aus.
6. Wählen Sie Attach Verified Access Trust Provider aus.

Trennen Sie einen Vertrauensanbieter von einer Instanz

Gehen Sie wie folgt vor, um einen Vertrauensanbieter von einer Verified Access-Instanz zu trennen.

So trennen Sie einen Vertrauensanbieter von einer Verified Access-Instanz

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances aus.
3. Wählen Sie die Instance aus.
4. Wählen Sie Aktionen, Vertrauensanbieter mit verifiziertem Zugriff trennen aus.
5. Wählen Sie für Verified Access Trust Provider den Trust Provider aus.
6. Wählen Sie Detach Verified Access Trust Provider aus.

Löschen Sie eine Instanz mit verifiziertem Zugriff

Wenn Sie mit einer Verified Access-Instanz fertig sind, können Sie sie löschen. Bevor Sie eine Instanz löschen können, müssen Sie alle zugehörigen Trust Provider oder Verified Access-Gruppen entfernen.

Um eine Verified Access-Instanz zu löschen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

2. Wählen Sie im Navigationsbereich Verified Access-Instanzen aus.
3. Wählen Sie die Verified Access-Instanz aus.
4. Wählen Sie Aktionen, Instanz mit verifiziertem Zugriff löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein und wählen Sie dann Delete (Löschen) aus.

Integration mit AWS WAF

Zusätzlich zu den Authentifizierungs- und Autorisierungsregeln, die von Verified Access durchgesetzt werden, möchten Sie möglicherweise auch einen Perimeterschutz anwenden. Dies kann Ihnen helfen, Ihre Anwendungen vor zusätzlichen Bedrohungen zu schützen. Sie können dies erreichen, indem Sie es AWS WAF in Ihre Verified Access-Bereitstellung integrieren. AWS WAFist eine Firewall für Webanwendungen, mit der Sie die HTTP (S) -Anfragen überwachen können, die an Ihre geschützten Webanwendungsressourcen weitergeleitet werden. Weitere Informationen zu AWS WAF finden Sie unter AWS WAF im AWS WAF-Entwickler-Leitfaden.

Sie können Verified Access AWS WAF integrieren, indem Sie einer Verified Access-Instanz eine AWS WAF Web Access Control List (ACL) zuordnen. Eine Web-ACL ist eine AWS WAF Ressource, die Ihnen eine genaue Kontrolle über alle HTTP (S) -Webanfragen gibt, auf die Ihre geschützte Ressource reagiert. Während der Bearbeitung AWS WAF der Zuordnungs- oder Trennungsanfrage wird der Status aller mit der Instance verbundenen Verified Access-Endpunkte als angezeigt. `updating` Nachdem die Anfrage abgeschlossen ist, kehrt der Status zu zurück. `active` Sie können den Status im AWS Management Console oder anzeigen, indem Sie den Endpunkt mit dem beschreibenAWS CLI.

> **Note**
>
> Sie können diese Integration auch AWS WAF über die Konsole oder API durchführen. Sie benötigen den Amazon Resource Name (ARN) Ihrer Verified Access-Instance. Sie können diesen ARN im folgenden Format erstellen:`arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}`.

Themen

- Für die Integration sind IAM-Berechtigungen erforderlich AWS WAF
- Ordnen Sie eine AWS WAF Web-ACL zu

- Überprüfen Sie den Status der Integration AWS WAF
- Trennen Sie die Zuordnung einer AWS WAF Web-ACL

Für die Integration sind IAM-Berechtigungen erforderlich AWS WAF

Die Integration AWS WAF mit Verified Access umfasst Aktionen, die nur auf Berechtigungen beschränkt sind und nicht direkt einem API-Vorgang entsprechen. Diese Aktionen sind in der AWS Identity and Access Management Serviceautorisierungsreferenz mit angegeben. `[permission only]` Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EC2 in der Service Authorization Reference.

Um mit einer Web-ACL arbeiten zu können, muss Ihr AWS Identity and Access Management Principal über die folgenden Berechtigungen verfügen.

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

Ordnen Sie eine AWS WAF Web-ACL zu

Die folgenden Schritte zeigen, wie Sie eine AWS WAF Web Access Control List (ACL) mit einer Verified Access-Instanz verknüpfen, indem Sie die verwendenAWS Management Console.

> **Tip**
>
> Sie benötigen eine bestehende AWS WAF Web-ACL, um das unten stehende Verfahren durchführen zu können. Weitere Informationen zu Web-ACLs finden Sie unter Web Access Control Lists im AWS WAFDeveloper Guide.

So ordnen Sie einer Verified Access-Instanz eine AWS WAF Web-ACL zu

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances aus.
3. Wählen Sie die Verified Access-Instanz aus.

4. Wählen Sie den Tab Integrationen aus.
5. Wählen Sie „Aktionen“ und anschließend „Web-ACL zuordnen“.
6. Wählen Sie für Web-ACL eine bestehende Web-ACL und dann Associate Web ACL aus.

Sie können diese Aufgabe auch AWS WAF mit dem Befehl AWS Management Console for ausführen. Weitere Informationen finden Sie unter Zuordnen oder Aufheben der Zuordnung einer Web-ACL zu einer AWS-Ressource im AWS WAF Entwicklerhandbuch.

Überprüfen Sie den Status der Integration AWS WAF

Mithilfe von können Sie überprüfen, ob eine AWS WAF Web Access Control List (ACL) einer Verified Access-Instanz zugeordnet ist oder nichtAWS Management Console.

Um den Status der AWS WAF Integration mit einer Verified Access-Instanz anzuzeigen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances aus.
3. Wählen Sie die Verified Access-Instanz aus.
4. Wählen Sie den Tab Integrationen aus.
5. Überprüfen Sie die Details, die unter WAF-Integrationsstatus aufgeführt sind. Der Status wird zusammen mit der Web-ACL-Kennung als Zugeordnet oder Nicht verknüpft angezeigt, sofern der Status Zugeordnet ist.

Trennen Sie die Zuordnung einer AWS WAF Web-ACL

Die folgenden Schritte zeigen, wie Sie die Zuordnung einer AWS WAF Web Access Control List (ACL) zu einer Verified Access-Instanz mithilfe von aufheben. AWS Management Console

So trennen Sie die Zuordnung einer AWS WAF Web-ACL zu einer Verified Access-Instanz

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances aus.
3. Wählen Sie die Verified Access-Instanz aus.
4. Wählen Sie den Tab Integrationen aus.
5. Wählen Sie „Aktionen“ und dann „Web-ACL trennen“.
6. Bestätigen Sie, indem Sie „Web-ACL trennen“ wählen.

Sie können diese Aufgabe auch mit dem AWS WAF Symbol AWS Management Console für ausführen. Weitere Informationen finden Sie unter Zuordnen oder Aufheben der Zuordnung einer Web-ACL zu einer AWS-Ressource im AWS WAF Entwicklerhandbuch.

FIPS-Konformität für verifizierten Zugriff

Der Federal Information Processing Standard (FIPS) ist ein US-amerikanischer und kanadischer Regierungsstandard, der Sicherheitsanforderungen für kryptografische Module zum Schutz vertraulicher Informationen festlegt. AWS Verified Accessbietet die Möglichkeit, Ihre Umgebung so zu konfigurieren, dass sie der FIPS-Publikation 140-2 entspricht. Die FIPS-Konformität für Verified Access ist in den folgenden Regionen verfügbar: AWS

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Kanada (Zentral)

Auf dieser Seite erfahren Sie, wie Sie eine neue oder eine bestehende Verified Access-Umgebung so konfigurieren, dass sie FIPS-konform ist.

Themen

Konfigurieren Sie eine bestehende Verified Access-Umgebung für die FIPS-Konformität

Wenn Sie über eine bestehende Verified Access-Umgebung verfügen und diese so konfigurieren möchten, dass sie FIPS-konform ist, müssen einige Ressourcen gelöscht und neu erstellt werden, um die FIPS-Konformität zu aktivieren.

Gehen Sie wie folgt vor, um eine bestehende AWS Verified Access Umgebung so zu konfigurieren, dass sie FIPS-konform ist.

1. Löschen Sie Ihre ursprünglichen Verified Access-Endpunkte, Gruppen und Instanzen. Ihre konfigurierten Vertrauensanbieter können wiederverwendet werden.
2. Erstellen Sie eine Verified Access-Instanz und achten Sie darauf, dass bei der Erstellung die Federal Information Process Standards (FIPS) aktiviert sind. Fügen Sie außerdem während der Erstellung den Verified Access-Vertrauensanbieter hinzu, den Sie verwenden möchten, indem Sie ihn aus der Dropdownliste auswählen.
3. Erstellen Sie eine Gruppe mit verifiziertem Zugriff. Während der Erstellung der Gruppe ordnen Sie sie der soeben erstellten Verified Access-Instanz zu.
4. Erstellen Sie eine oder mehrereVerifizierte Zugriffsendpunkte. Bei der Erstellung Ihrer Endpunkte ordnen Sie sie der Gruppe zu, die Sie im vorherigen Schritt erstellt haben.

Konfigurieren Sie eine neue Verified Access-Umgebung für die FIPS-Konformität

Gehen Sie wie folgt vor, um eine neue AWS Verified Access Umgebung zu konfigurieren, die FIPS-konform ist.

1. Konfigurieren Sie einen Vertrauensanbieter. Je nach Ihren Anforderungen müssen Sie einen Vertrauensanbieter für Benutzeridentitäten und (optional) einen gerätebasierten Vertrauensanbieter einrichten.
2. Erstellen Sie eine Verified Access-Instanz und achten Sie darauf, dass Sie während des Vorgangs die Federal Information Process Standards (FIPS) aktivieren. Fügen Sie bei der Erstellung außerdem den Verified Access-Vertrauensanbieter hinzu, den Sie im vorherigen Schritt erstellt haben, indem Sie ihn aus der Dropdownliste auswählen.
3. Erstellen Sie eine Verified Access-Gruppe. Während der Erstellung der Gruppe ordnen Sie sie der soeben erstellten Verified Access-Instanz zu.
4. Erstellen Sie eine oder mehrereVerifizierte Zugriffsendpunkte. Bei der Erstellung Ihrer Endpunkte ordnen Sie sie der Gruppe zu, die Sie im vorherigen Schritt erstellt haben.

Vertraue Anbietern für verifizierten Zugriff

Ein Vertrauensanbieter ist ein Dienst, der Informationen über Benutzer und Geräte an AWS Verified Access sendet. Diese Informationen werden als Vertrauenskontext bezeichnet. Dazu können Attribute gehören, die auf der Benutzeridentität basieren, z. B. eine E-Mail-Adresse oder die Mitgliedschaft in der „Verkaufsorganisation“, oder Geräteinformationen wie installierte Sicherheitspatches oder die Version der Antivirensoftware.

Verified Access unterstützt die folgenden Kategorien von Vertrauensanbietern:

- Benutzeridentität — Ein Identitätsanbieterdienst (IdP), der digitale Identitäten für Benutzer speichert und verwaltet.
- Geräteverwaltung — Ein Geräteverwaltungssystem für Geräte wie Laptops, Tablets und Smartphones.

Inhalt

Vertrauensanbieter für Benutzeridentitäten

Sie können wählen, ob Sie AWS IAM Identity Center entweder einen OpenID Connect-kompatiblen Vertrauensanbieter für Benutzeridentitäten verwenden möchten.

Inhalt

Verwenden Sie IAM Identity Center als Vertrauensanbieter

Sie können es AWS IAM Identity Center als Vertrauensanbieter für Benutzeridentitäten mit AWS verifiziertem Zugriff verwenden.

Voraussetzungen und Überlegungen

- Ihre IAM Identity Center-Instanz muss eine Instanz AWS Organizations sein. Eine IAM Identity Center-Instanz mit einem eigenständigen AWS Konto funktioniert nicht.
- Ihre IAM Identity Center-Instanz muss in derselben AWS Region aktiviert sein, in der Sie den Verified Access Trust Provider erstellen möchten.

Einzelheiten zu den verschiedenen Instanztypen finden Sie im AWS IAM Identity CenterBenutzerhandbuch unter Organisations- und Kontoinstanzen von IAM Identity Center verwalten.

Erstellen Sie einen IAM Identity Center Trust Provider

Nachdem IAM Identity Center für Ihr AWS Konto aktiviert wurde, können Sie das folgende Verfahren verwenden, um IAM Identity Center als Ihren Vertrauensanbieter für verifizierten Zugriff einzurichten.

So erstellen Sie einen IAM Identity Center-Vertrauensanbieter (Konsole) AWS

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access Trust Provider und dann Create Verified Access Trust Provider aus.
3. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Vertrauensanbieter ein.
4. Geben Sie als Referenzname für die Richtlinie einen Bezeichner ein, der später bei der Arbeit mit Richtlinienregeln verwendet werden soll.
5. Wählen Sie unter Vertrauensanbietertyp die Option Benutzervertrauensdienstanbieter aus.
6. Wählen Sie unter Benutzer-Trust-Provider-Typ die Option IAM Identity Center aus.
7. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
8. Wählen Sie Create Verified Access Trust Provider.

So erstellen Sie einen IAM Identity Center Trust Provider (AWSCLI)

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

Löschen Sie einen IAM Identity Center-Vertrauensanbieter

Bevor Sie einen Trust Provider löschen können, müssen Sie die gesamte Endpoint- und Gruppenkonfiguration aus der Instance entfernen, an die der Trust Provider angehängt ist.

Um einen IAM Identity Center Trust Provider (AWSKonsole) zu löschen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access Trust Provider und dann unter Verified Access Trust Providers den Trust Provider aus, den Sie löschen möchten.
3. Wählen Sie Aktionen und dann Vertrauensanbieter mit verifiziertem Zugriff löschen aus.
4. Bestätigen Sie den Löschvorgang, indem Sie die Eingabe `delete` in das Textfeld eingeben.
5. Wählen Sie Löschen aus.

So löschen Sie einen IAM Identity Center Trust Provider (AWSCLI)

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

Verwendung eines OpenID Connect-Vertrauensanbieters

AWSVerified Access unterstützt Identitätsanbieter, die standardmäßige OpenID Connect (OIDC) -Methoden verwenden. Sie können OIDC-kompatible Anbieter als Vertrauensanbieter für Benutzeridentitäten mit Verified Access verwenden. Aufgrund der Vielzahl potenzieller OIDC-Anbieter AWS ist es jedoch nicht möglich, jede OIDC-Integration mit Verified Access zu testen.

Verified Access bezieht die Vertrauensdaten, die es auswertet, von den OIDC-Anbietern. `UserInfo Endpoint` Der `Scope` Parameter wird verwendet, um zu bestimmen, welche Vertrauensdatensätze abgerufen werden. Nachdem die Vertrauensdaten empfangen wurden, wird die Verified Access-Richtlinie anhand dieser Daten bewertet.

> **Note**
>
> Verified Access verwendet bei der Bewertung der Verified Access-Richtlinie keine Vertrauensdaten aus den vom OIDC-Anbieter `ID token` gesendeten Daten. Nur vertrauenswürdige Daten von werden anhand `UserInfo Endpoint` der Richtlinie bewertet.

Inhalt

Voraussetzungen für die Erstellung eines OIDC-Vertrauensanbieters

Sie müssen die folgenden Informationen direkt von Ihrem Trust Provider-Dienst einholen:

- Aussteller
- Endpunkt der Autorisierung
- Token-Endpunkt
- UserInfo Endpunkt
- Client-ID
- Clientschlüssel
- Scope

Erstellen Sie einen OIDC-Vertrauensanbieter

Gehen Sie wie folgt vor, um einen OIDC als Ihren Vertrauensanbieter zu erstellen.

So erstellen Sie einen OIDC-Vertrauensanbieter (Konsole) AWS

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Vertrauensanbieter mit verifiziertem Zugriff und dann Vertrauensanbieter mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Vertrauensanbieter ein.
4. Geben Sie als Referenzname für die Richtlinie einen Bezeichner ein, der später bei der Arbeit mit Richtlinienregeln verwendet werden soll.
5. Wählen Sie unter Vertrauensanbietertyp die Option Benutzervertrauensdienstanbieter aus.
6. Wählen Sie unter Benutzervertrauensanbietertyp die Option OIDC (OpenID Connect) aus.
7. Geben Sie für Issuer die ID des OIDC-Emittenten ein.

8. Geben Sie für Autorisierungsendpunkt die vollständige URL des Autorisierungsendpunkts ein.
9. Geben Sie für Token-Endpunkt die vollständige URL des Token-Endpunkts ein.
10. Geben Sie für Benutzerendpunkt die vollständige URL des Benutzerendpunkts ein.
11. Geben Sie die OAuth 2.0-Client-ID als Client-ID ein.
12. Geben Sie den geheimen OAuth 2.0-Clientschlüssel für Client-Schlüssel ein.
13. Geben Sie eine durch Leerzeichen getrennte Liste von Bereichen ein, die mit Ihrem Identitätsanbieter definiert wurden. Für Scope ist mindestens der Bereich „openid“ erforderlich.
14. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
15. Wählen Sie Create Verified Access Trust Provider aus.

> **Note**
>
> Sie müssen der Zulassungsliste Ihres OIDC-Anbieters eine Weiterleitungs-URI hinzufügen. Zu diesem Zweck sollten Sie den `ApplicationDomain` Endpunkt „Verified Access“ verwenden. Dies finden Sie in derAWS Management Console, auf der Registerkarte Details für Ihren verifizierten Zugriffs-Endpunkt oder indem Sie AWS CLI den Endpunkt beschreiben. Fügen Sie Folgendes zur Zulassungsliste Ihres OIDC-Anbieters hinzu: https:///oauth2/idpresponse `ApplicationDomain`

So erstellen Sie einen OIDC Trust Provider (CLI) AWS

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

Ändern Sie einen OIDC-Vertrauensanbieter

Nachdem Sie einen Vertrauensanbieter erstellt haben, können Sie dessen Konfiguration aktualisieren.

Um einen OIDC-Vertrauensanbieter (AWSKonsole) zu ändern

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Trust Providers aus, und wählen Sie dann unter Verified Access Trust Providers den Vertrauensanbieter aus, den Sie ändern möchten.

3. Wählen Sie Aktionen und dann Vertrauensanbieter mit verifiziertem Zugriff ändern aus.
4. Ändern Sie die Optionen, die Sie ändern möchten.
5. Wählen Sie Vertrauensanbieter mit verifiziertem Zugriff ändern aus.

So ändern Sie einen OIDC-Vertrauensanbieter (CLI) AWS

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

Löschen Sie einen OIDC-Vertrauensanbieter

Bevor Sie einen Benutzer-Vertrauensanbieter löschen können, müssen Sie zunächst die gesamte Endpunkt- und Gruppenkonfiguration aus der Instanz entfernen, an die der Trust Provider angehängt ist.

Um einen OIDC-Vertrauensanbieter (AWSKonsole) zu löschen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Trust Providers aus, und wählen Sie dann unter Verified Access Trust Providers den Vertrauensanbieter aus, den Sie löschen möchten.
3. Wählen Sie Aktionen und dann Vertrauensanbieter mit verifiziertem Zugriff löschen aus.
4. Bestätigen Sie den Löschvorgang, indem Sie die Eingabe `delete` in das Textfeld eingeben.
5. Wählen Sie Löschen aus.

So löschen Sie einen OIDC-Vertrauensanbieter (CLI) AWS

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

Gerätebasierte Vertrauensanbieter

Sie können vertrauenswürdige Geräteanbieter mit AWS verifiziertem Zugriff verwenden. Sie können einen oder mehrere Vertrauensanbieter für Geräte mit Ihrer Verified Access-Instanz verwenden.

Inhalt

- Erstellen Sie einen gerätebasierten Vertrauensanbieter
- Ändern Sie einen gerätebasierten Vertrauensanbieter
- Löscht einen gerätebasierten Vertrauensanbieter

Unterstützte Anbieter von Gerätevertrauensstellungen

Die folgenden Anbieter von Gerätevertrauen können in Verified Access integriert werden:

- CrowdStrike — Sicherung privater Anwendungen mit CrowdStrike verifiziertem Zugriff
- Jamf — Integration von verifiziertem Zugriff mit Jamf Device Identity
- JumpCloud — Integration JumpCloud und verifizierter Zugriff AWS

Erstellen Sie einen gerätebasierten Vertrauensanbieter

Gehen Sie wie folgt vor, um einen Gerätevertrauensanbieter für die Verwendung mit Verified Access zu erstellen und zu konfigurieren.

So erstellen Sie einen Vertrauensdienstanbieter für Geräte mit verifiziertem Zugriff (AWSKonsole)

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Vertrauensanbieter mit verifiziertem Zugriff und dann Vertrauensanbieter mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Vertrauensanbieter ein.
4. Geben Sie einen Bezeichner ein, der später bei der Arbeit mit Richtlinienregeln für den Referenznamen verwendet werden soll.
5. Wählen Sie als Vertrauensanbietertyp die Option Geräteidentität aus.
6. Wählen Sie als Typ der Geräteidentität die Option Jamf, CrowdStrike, oder JumpCloudaus.
7. Geben Sie unter Mandanten-ID den Bezeichner der Mandantenanwendung ein.
8. (Optional) Geben Sie für die URL des öffentlichen Signaturschlüssels die eindeutige Schlüssel-URL ein, die Ihnen von Ihrem Device Trust Provider zur Verfügung gestellt wurde. (Dieser Parameter ist für Jamf CrowdStrike oder Jumpcloud nicht erforderlich.)
9. Wählen Sie Create Verified Access Trust Provider aus.

> **Note**
>
> Sie müssen der Zulassungsliste Ihres OIDC-Anbieters eine Weiterleitungs-URI hinzufügen. Zu diesem Zweck sollten Sie den `DeviceValidationDomain` Endpunkt „Verified Access“ verwenden. Dies finden Sie in derAWS Management Console, auf der Registerkarte Details für Ihren verifizierten Zugriffs-Endpunkt oder indem Sie AWS CLI den Endpunkt beschreiben. Fügen Sie Folgendes zur Zulassungsliste Ihres OIDC-Anbieters hinzu: https:///oauth2/idpresponse `DeviceValidationDomain`

So erstellen Sie einen Device Trust Provider (AWSCLI) mit verifiziertem Zugriff

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

Ändern Sie einen gerätebasierten Vertrauensanbieter

Nachdem Sie einen Vertrauensanbieter erstellt haben, können Sie dessen Konfiguration aktualisieren.

So ändern Sie einen Vertrauensdienstanbieter für Geräte mit verifiziertem Zugriff (AWSKonsole)

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Trust Providers aus.
3. Wählen Sie den Vertrauensanbieter aus.
4. Wählen Sie Aktionen und dann Vertrauensanbieter mit verifiziertem Zugriff ändern aus.
5. Ändern Sie die Beschreibung nach Bedarf.
6. (Optional) Ändern Sie für die URL des öffentlichen Signaturschlüssels die eindeutige Schlüssel-URL, die Ihnen von Ihrem Device Trust Provider zur Verfügung gestellt wurde. (Dieser Parameter ist nicht erforderlich, wenn es sich bei Ihrem Gerät um Jamf CrowdStrike oder Jumpcloud handelt.)
7. Wählen Sie Vertrauensanbieter mit verifiziertem Zugriff ändern aus.

So ändern Sie einen Device Trust Provider (AWSCLI) mit verifiziertem Zugriff

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

Löscht einen gerätebasierten Vertrauensanbieter

Wenn Sie mit einem Vertrauensanbieter fertig sind, können Sie ihn löschen.

So löschen Sie einen Vertrauensanbieter für Geräte mit verifiziertem Zugriff (AWSKonsole)

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Trust Providers aus.
3. Wählen Sie unter Vertrauensanbieter mit verifiziertem Zugriff den Vertrauensanbieter aus, den Sie löschen möchten.
4. Wählen Sie Aktionen und dann Vertrauensanbieter mit verifiziertem Zugriff löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein und wählen Sie dann Delete (Löschen) aus.

So löschen Sie einen Device Trust Provider (AWSCLI) mit verifiziertem Zugriff

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

Gruppen mit verifiziertem Zugriff

Eine AWS Verified Access-Gruppe ist eine Sammlung von Verified Access-Endpunkten und eine Verified Access-Richtlinie auf Gruppenebene. Jeder Endpunkt innerhalb einer Gruppe verwendet die Richtlinien für verifizierten Zugriff. Sie können Gruppen verwenden, um Endpunkte mit gemeinsamen Sicherheitsanforderungen zusammenzustellen. Dies kann dazu beitragen, die Richtlinienverwaltung zu vereinfachen, indem eine Richtlinie für die Sicherheitsanforderungen mehrerer Anwendungen verwendet wird.

Sie können beispielsweise alle Vertriebsanwendungen gruppieren und eine gruppenweite Zugriffsrichtlinie festlegen. Sie können diese Richtlinie dann verwenden, um gemeinsame Mindestsicherheitsanforderungen für alle Vertriebsanwendungen zu definieren. Dieser Ansatz trägt zur Vereinfachung der politischen Verwaltung bei.

Wenn Sie eine Gruppe mit verifizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfiz Während der Erstellung eines Endpoints ordnen Sie den Endpunkt einer Gruppe zu.

Aufgaben

Gruppe mit verifizfizfizfizfizfizfizfiz

Gehen Sie wie folgt vor, um eine Gruppe verifizfizfizfizfizfizfizfizfizfizfizfizfiz

So verifizfizfizfizfizfizfizfizfizfizfiz

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Gruppen und dann Verified Access-Gruppe erstellen aus.
3. (Optional) Geben Sie für Namensschild und Beschreibung einen Namen und eine Beschreibung für die Gruppe ein.
4. Wählen Sie für die Verified Access-Instanz eine Verified Access-Instanz aus, die Sie der Gruppe zuordnen möchten.

5. (Optional) Geben Sie für die Richtliniendefinition eine Richtlinie für verifizierten Zugriff ein, die auf die Gruppe angewendet werden soll.
6. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
7. Wählen Sie Gruppe „Verifizierten Zugriff erstellen“ aus.

Eine Gruppenverifizfizfizfizfizfizfizfizfizfiz

Gehen Sie wie folgt vor, um eine Gruppenverifizfizfizfizfizfizfizfizfizfizfizfizfizfizfiz

So modifizieren Sie eine Gruppenverifizfizfizfizfizfizfizfiz

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich verifizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfiz
3. Wählen Sie Aktionen und dann Gruppenrichtlinie für verifizierten Zugriff ändern aus.
4. (Optional) Aktivieren oder deaktivieren Sie die Richtlinie aktivieren, je nachdem, welches Ziel Sie gerade verfolgen.
5. (Optional) Geben Sie für Richtlinie eine Richtlinie für verifizierten Zugriff ein, die auf die Gruppe angewendet werden soll.
6. Wählen Sie „Gruppenrichtlinie für verifizierten Zugriff ändern“.

Eine Gruppe verifizfizfizfizfizfizfizfiz

Wenn Sie eine Gruppe verifizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfiz

So verifizfizfizfizfizfizfizfizfizfizfiz

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich verifizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfizfiz
3. Wählen Sie die -Gruppe aus.
4. Wählen Sie Aktionen, Gruppe „Verifizierten Zugriff löschen“.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein und wählen Sie dann Delete (Löschen) aus.

Verifizierte Zugriffsendpunkte

Ein Verified Access-Endpunkt steht für eine Anwendung. Jeder Endpunkt ist einer Verified Access-Gruppe zugeordnet und erbt die Zugriffsrichtlinie für die Gruppe. Sie können optional jedem Endpunkt eine anwendungsspezifische Endpunktrichtlinie zuordnen.

Inhalt

Verifizierte Access-Endpunkttypen

Im Folgenden sind die möglichen Endpunkttypen aufgeführt:

- Load Balancer — Anwendungsanfragen werden an einen Load Balancer gesendet, um sie an Ihre Anwendung zu verteilen.
- Netzwerkschnittstelle — Anwendungsanfragen werden unter Verwendung des angegebenen Protokolls und Ports an eine Netzwerkschnittstelle gesendet.

Gemeinsam genutzte VPCs und Subnetze

Im Folgenden sind die Verhaltensweisen in Bezug auf gemeinsam genutzte VPC-Subnetze aufgeführt:

- Verified Access-Endpunkte werden durch die gemeinsame Nutzung von VPC-Subnetzen unterstützt. Ein Teilnehmer kann einen Verified Access-Endpunkt in einem gemeinsam genutzten Subnetz erstellen.

- Der Teilnehmer, der den Endpunkt erstellt hat, ist der Besitzer des Endpunkts und die einzige Partei, die den Endpunkt ändern darf. Der VPC-Besitzer darf den Endpunkt nicht ändern.
- Verifizierte Zugriffsendpunkte können nicht in einer AWS lokalen Zone erstellt werden, weshalb eine gemeinsame Nutzung über Local Zones nicht möglich ist.

Weitere Informationen finden Sie unter Freigeben Ihrer VPC für andere Konten im Amazon-VPC-Benutzerhandbuch.

Erstellen Sie einen Load Balancer-Endpunkt für verifizierten Zugriff

Gehen Sie wie folgt vor, um einen Load Balancer-Endpunkt zu erstellen. Weitere Informationen zu Load Balancern finden Sie im Elastic Load Balancing User Guide.

Voraussetzungen

- Es wird nur IPv4-Verkehr unterstützt.
- Nur die Protokolle HTTP und HTTPS werden unterstützt.
- Der Load Balancer muss entweder ein Application Load Balancer oder ein Network Load Balancer sein, und es muss sich um einen internen Load Balancer handeln.
- Der Load Balancer und die Subnetze müssen zu derselben Virtual Private Cloud (VPC) gehören.
- HTTPS-Load Balancer können entweder selbstsignierte oder öffentliche TLS-Zertifikate verwenden.
- Sie müssen einen Domainnamen für Ihre Anwendung angeben. Dies ist der öffentliche DNS-Name, den Ihre Benutzer für den Zugriff auf Ihre Anwendung verwenden werden. Sie müssen außerdem ein öffentliches SSL-Zertifikat mit einer CN bereitstellen, die diesem Domainnamen entspricht. Sie können das Zertifikat mit erstellen oder importierenAWS Certificate Manager.

Um einen Load Balancer-Endpunkt zu erstellen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access Endpoints aus.
3. Wählen Sie Endpunkt mit verifiziertem Zugriff erstellen aus.
4. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Endpunkt ein.

5. Wählen Sie für Gruppe mit verifiziertem Zugriff eine Gruppe mit verifiziertem Zugriff für den Endpunkt aus.
6. Gehen Sie wie folgt vor, um Anwendungsdetails zu erhalten:
   a. Geben Sie unter Anwendungsdomäne einen DNS-Namen für Ihre Anwendung ein.
   b. Wählen Sie unter Domainzertifikat ARN das öffentliche TLS-Zertifikat aus.
7. Gehen Sie wie folgt vor, um Endpunktdetails zu erhalten:
   a. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.
   b. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppen für den Endpunkt aus. Der Datenverkehr vom Verified Access-Endpunkt, der in Ihren Load Balancer gelangt, wird dieser Sicherheitsgruppe zugeordnet.
   c. Geben Sie für das Endpoint-Domänenpräfix einen benutzerdefinierten Bezeichner ein, der dem DNS-Namen vorangestellt wird, den Verified Access für den Endpunkt generiert.
   d. Wählen Sie als Endpunkttyp die Option Load Balancer aus.
   e. Wählen Sie als Protokoll HTTPS oder HTTP aus.
   f. Geben Sie unter Port die Portnummer ein.
   g. Wählen Sie für Load Balancer ARN den Load Balancer aus.
   h. Wählen Sie für Subnetze die Subnetze für Ihren Load Balancer aus.
8. (Optional) Geben Sie für die Richtliniendefinition eine Richtlinie für verifizierten Zugriff für den Endpunkt ein.
9. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
10. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Erstellen Sie einen Netzwerkschnittstellen-Endpunkt für Verified Access

Gehen Sie wie folgt vor, um einen Netzwerkschnittstellen-Endpunkt zu erstellen.

Voraussetzungen

- Nur IPv4-Verkehr wird unterstützt.
- Nur die Protokolle HTTP und HTTPS werden unterstützt.

- Die Netzwerkschnittstelle muss zu derselben Virtual Private Cloud (VPC) gehören wie die Sicherheitsgruppen.
- Wir verwenden die private IP auf der Netzwerkschnittstelle, um den Verkehr weiterzuleiten.
- Sie müssen einen Domainnamen für Ihre Anwendung angeben. Dies ist der öffentliche DNS-Name, den Ihre Benutzer für den Zugriff auf Ihre Anwendung verwenden werden. Sie müssen außerdem ein öffentliches SSL-Zertifikat mit einer CN bereitstellen, die diesem Domainnamen entspricht. Sie können das Zertifikat mit erstellen oder importierenAWS Certificate Manager.

Um einen Netzwerkschnittstellen-Endpunkt zu erstellen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Endpoints aus.
3. Wählen Sie Endpunkt mit verifiziertem Zugriff erstellen aus.
4. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Endpunkt ein.
5. Wählen Sie für Gruppe mit verifiziertem Zugriff eine Gruppe mit verifiziertem Zugriff für den Endpunkt aus.
6. Gehen Sie wie folgt vor, um Anwendungsdetails zu erhalten:
   a. Geben Sie unter Anwendungsdomäne den DNS-Namen für Ihre Anwendung ein.
   b. Wählen Sie unter Domainzertifikat ARN das öffentliche TLS-Zertifikat aus.
7. Gehen Sie wie folgt vor, um Endpunktdetails zu erhalten:
   a. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.
   b. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppen für den Endpunkt aus. Datenverkehr vom Verified Access-Endpunkt, der in Ihre Netzwerkschnittstelle gelangt, wird dieser Sicherheitsgruppe zugeordnet.
   c. Geben Sie für das Endpunktdomänenpräfix eine benutzerdefinierte Kennung ein, die dem DNS-Namen vorangestellt wird, den Verified Access für den Endpunkt generiert.
   d. Wählen Sie als Endpunkttyp die Option Netzwerkschnittstelle aus.
   e. Wählen Sie als Protokoll HTTPS oder HTTP aus.
   f. Geben Sie unter Port die Portnummer ein.
   g. Wählen Sie für Netzwerkschnittstelle die Netzwerkschnittstelle aus.

8. (Optional) Geben Sie für die Richtliniendefinition eine Richtlinie für verifizierten Zugriff für den Endpunkt ein.
9. (Optional) Sie fügen ein Tag hinzu, indem Sie neuen Tag hinzufügen auswählen und den Schlüssel und den Wert für den Tag eingeben.
10. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Lassen Sie Datenverkehr zu, der von Ihrem Verified Access-Endpunkt stammt

Sie können die Sicherheitsgruppen für Ihre Anwendungen so konfigurieren, dass sie Datenverkehr zulassen, der von Ihrem Verified Access-Endpunkt stammt. Dazu fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die die Sicherheitsgruppe für den Endpunkt als Quelle angibt. Wir empfehlen, dass Sie alle zusätzlichen Regeln für eingehenden Datenverkehr entfernen, sodass Ihre Anwendung nur Datenverkehr von Ihrem Verified Access-Endpunkt empfängt.

Wir empfehlen Ihnen, Ihre bestehenden Regeln für ausgehenden Datenverkehr beizubehalten.

Um die Sicherheitsgruppenregeln für Ihre Anwendung zu aktualisieren

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Endpoints aus.
3. Wählen Sie den Endpunkt Verified Access aus, suchen Sie auf der Registerkarte Details nach Sicherheitsgruppen-IDs und kopieren Sie die ID der Sicherheitsgruppe für Ihren Endpunkt.
4. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
5. Aktivieren Sie das Kontrollkästchen für die Sicherheitsgruppe, die Ihrem Ziel zugeordnet ist, und wählen Sie dann Aktionen, Regeln für eingehenden Datenverkehr bearbeiten aus.
6. Gehen Sie wie folgt vor, um eine Sicherheitsgruppenregel hinzuzufügen, die Datenverkehr zulässt, der von Ihrem Verified Access-Endpunkt stammt:
    a. Wählen Sie Add rule.
    b. Wählen Sie unter Typ die Option Gesamter Verkehr oder den spezifischen Datenverkehr aus, der zugelassen werden soll.
    c. Wählen Sie für Quelle die Option Benutzerdefiniert aus und fügen Sie die ID der Sicherheitsgruppe für Ihren Endpunkt ein.

7. (Optional) Wenn Sie festlegen möchten, dass der Datenverkehr nur von Ihrem Verified Access-Endpunkt stammt, löschen Sie alle anderen Sicherheitsgruppenregeln für eingehenden Datenverkehr.
8. Wählen Sie Save rules (Regeln speichern) aus.

Ändern Sie einen Endpunkt mit verifiziertem Zugriff

Nachdem Sie einen Verified Access-Endpunkt erstellt haben, können Sie dessen Konfiguration aktualisieren.

Um einen Verified Access-Endpunkt zu ändern

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Endpoints aus.
3. Wählen Sie den Endpunkt.
4. Wählen Sie Aktionen, Endpunkt für verifizierten Zugriff ändern aus.
5. Ändern Sie die Endpunktdetails nach Bedarf.
6. Wählen Sie Endpunkt für verifizierten Zugriff ändern aus.

Ändern Sie eine Endpunktrichtlinie für verifizierten Zugriff

Nachdem Sie einen Verified Access-Endpunkt erstellt haben, können Sie dessen Richtlinie ändern.

Um eine Endpunktrichtlinie für verifizierten Zugriff zu ändern

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Endpoints aus.
3. Wählen Sie den Endpunkt aus, dessen Richtlinie Sie ändern möchten.
4. Wählen Sie Aktionen, Endpunktrichtlinie für verifizierten Zugriff ändern aus.
5. (Optional) Aktivieren oder deaktivieren Sie die Option „Richtlinie aktivieren“ je nach Ihrem aktuellen Ziel.
6. (Optional) Geben Sie unter Richtlinie eine Richtlinie für verifizierten Zugriff ein, die auf den Endpunkt angewendet werden soll.
7. Wählen Sie Endpunktrichtlinie für verifizierten Zugriff ändern aus.

Löschen Sie einen Endpunkt mit verifiziertem Zugriff

Wenn Sie mit einem Verified Access-Endpunkt fertig sind, können Sie ihn löschen.

Um einen Verified Access-Endpunkt zu löschen

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich die Option Verified Access Endpoints aus.
3. Wählen Sie den Endpunkt.
4. Wählen Sie Aktionen, Endpunkt mit verifiziertem Zugriff löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`delete`** ein und wählen Sie dann Löschen aus.

Vertraue Daten von Vertrauensanbietern

Vertrauensdaten sind Daten, die von einem Vertrauensanbieter an AWS Verified Access gesendet werden. Sie werden manchmal auch als „Benutzeransprüche“ oder „Vertrauenskontext“ bezeichnet. Die Daten enthalten im Allgemeinen Informationen über einen Benutzer oder ein Gerät. Beispiele für Vertrauensdaten sind die E-Mail-Adresse des Benutzers, die Gruppenmitgliedschaft, die Betriebssystemversion des Geräts, der Sicherheitsstatus des Geräts und mehr. Die gesendeten Informationen variieren je nach Vertrauensanbieter. Eine vollständige und aktualisierte Liste der Vertrauensdaten finden Sie daher in der Dokumentation Ihres Vertrauensanbieters.

Mithilfe der Protokollierungsfunktionen für verifizierten Zugriff können Sie jedoch auch sehen, welche Vertrauensdaten von Ihrem Vertrauensanbieter gesendet werden. Dies kann sehr nützlich sein, wenn Sie Richtlinien definieren, die den Zugriff auf Ihre Anwendungen zulassen oder verweigern. Informationen dazu, wie Sie Vertrauenskontext in Ihre Protokolle aufnehmen können, finden Sie unterEinschließen des Vertrauenskontexts.

Dieser Abschnitt enthält Beispiele für Vertrauensdaten und Beispiele für die ersten Schritte beim Schreiben von Richtlinien. Die hier bereitgestellten Informationen dienen nur zur Veranschaulichung und nicht als offizielle Referenz.

Inhalt

Standardkontext „Verifizierter Zugriff“

AWSVerified Access enthält standardmäßig einige Elemente zur aktuellen HTTP-Anfrage in allen Cedar-Evaluierungen, unabhängig von Ihren konfigurierten Vertrauensanbietern. Wenn eine Richtlinie bewertet wird, enthält Verified Access Daten über die aktuelle HTTP-Anfrage im Cedar-Kontext unter`context.http_request key`. Wenn Sie möchten, können Sie eine Richtlinie schreiben, die anhand der Daten bewertet wird. Das folgende JSON-Schema zeigt, welche Daten in der Auswertung enthalten sind.

```
{
```

```
    "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"
        }
    }
}
```

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die anhand der HTTP-Anforderungsdaten ausgewertet wird.

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

AWS IAM Identity Center

Wenn eine Richtlinie ausgewertet wird und Sie sie AWS IAM Identity Center als Vertrauensanbieter definieren, schließt AWS Verified Access die Vertrauensdaten im Cedar-Kontext unter dem Schlüssel

ein, den Sie in der Trust-Provider-Konfiguration als „Policy Reference Name“ angeben. Wenn Sie möchten, können Sie eine Richtlinie schreiben, die anhand der Vertrauensdaten bewertet wird.

> **Note**
>
> Der Kontextschlüssel für Ihren Vertrauensanbieter stammt aus dem Referenznamen der Richtlinie, den Sie bei der Erstellung des Vertrauensanbieters konfigurieren. Wenn Sie den Referenznamen der Richtlinie beispielsweise als „idp123" konfigurieren, lautet der Kontextschlüssel „context.idp123". Vergewissern Sie sich, dass Sie den richtigen Kontextschlüssel verwenden, wenn Sie die Richtlinie erstellen.

Das folgende JSON-Schema zeigt, welche Daten in der Auswertung enthalten sind.

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

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die anhand der von AWS IAM Identity Center bereitgestellten Vertrauensdaten bewertet wird.

```
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**
>
> Da Gruppennamen geändert werden können, bezieht sich IAM Identity Center auf Gruppen, die ihre Gruppen-ID verwenden. Auf diese Weise wird vermieden, dass bei der Änderung des Gruppennamens gegen eine Richtlinienaussage verstoßen wird.

Vertrauenswürdige Drittanbieter

In diesem Abschnitt werden die Vertrauensdaten beschrieben, die AWS Verified Access von Drittanbietern zur Verfügung gestellt werden.

> **Note**
>
> Der Kontextschlüssel für Ihren Vertrauensanbieter stammt aus dem Referenznamen der Richtlinie, den Sie bei der Erstellung des Vertrauensanbieters konfigurieren. Wenn Sie den Referenznamen der Richtlinie beispielsweise als „idp123" konfigurieren, lautet der Kontextschlüssel „context.idp123". Stellen Sie sicher, dass Sie beim Erstellen der Richtlinie den richtigen Kontextschlüssel verwenden.

Inhalt

Browser-Erweiterung

Wenn Sie beabsichtigen, den Kontext der Gerätevertrauensstellung in Ihre Zugriffsrichtlinien zu integrieren, benötigen Sie entweder die Browsererweiterung AWS Verified Access oder die Browsererweiterung eines anderen Partners. Verified Access unterstützt derzeit die Browser Google Chrome und Mozilla Firefox.

Wir unterstützen derzeit drei vertrauenswürdige Anbieter für Geräte: Jamf (das macOS-Geräte unterstützt), CrowdStrike (das Windows 11- und Windows 10-Geräte unterstützt) und JumpCloud (das sowohl Windows als auch macOS unterstützt).

- Wenn Sie in Ihren Richtlinien vertrauenswürdige Daten von Jamf verwenden, müssen Ihre Benutzer die Browsererweiterung AWS Verified Access aus dem Chrome Web Store oder der Firefox Add-On-Website auf ihren Geräten herunterladen und installieren.
- Wenn Sie CrowdStrikeVertrauensdaten in Ihren Richtlinien verwenden, müssen Ihre Benutzer zunächst den AWSVerified Access Native Messaging Host (direkter Download-Link) installieren.

Diese Komponente ist erforderlich, um die Vertrauensdaten von dem CrowdStrike Agenten abzurufen, der auf den Geräten der Benutzer ausgeführt wird. Nach der Installation dieser Komponente müssen Benutzer dann die Browsererweiterung AWS Verified Access aus dem Chrome-Webshop oder der Firefox-Add-On-Website auf ihren Geräten installieren.

- Wenn Sie sie verwenden JumpCloud, müssen Ihre Nutzer die JumpCloud Browsererweiterung aus dem Chrome-Webshop oder der Firefox-Add-On-Website auf ihren Geräten installiert haben.

Jamf

Jamf ist ein vertrauenswürdiger Drittanbieter. Wenn Sie Jamf bei der Bewertung einer Richtlinie als Vertrauensanbieter definieren, schließt Verified Access die Vertrauensdaten im Cedar-Kontext unter dem Schlüssel ein, den Sie in der Trust-Provider-Konfiguration als „Policy Reference Name“ angeben. Wenn Sie möchten, können Sie eine Richtlinie schreiben, die anhand der Vertrauensdaten bewertet wird. Das folgende JSON-Schema zeigt, welche Daten in der Bewertung enthalten sind.

Weitere Informationen zur Verwendung von Jamf with AWS Verified Access finden Sie unter Integrating AWS Verified Access with Jamf Device Identity auf der Jamf-Website.

```
{
    "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)"
        }
    }
}
```

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die anhand der von Jamf bereitgestellten Vertrauensdaten bewertet wird.

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

Cedar bietet eine nützliche `.contains()` Funktion, die bei Aufzählungen wie dem Risiko-Score von Jamf hilft.

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

CrowdStrike

CrowdStrike ist ein vertrauenswürdiger Drittanbieter. Wenn Sie eine Richtlinie CrowdStrike als Vertrauensanbieter definieren, schließt Verified Access die Vertrauensdaten im Cedar-Kontext unter dem Schlüssel ein, den Sie in der Trust-Provider-Konfiguration als „Richtlinien-Referenzname“ angeben. Wenn Sie möchten, können Sie eine Richtlinie schreiben, die anhand der Vertrauensdaten bewertet wird. Das folgende JSON-Schema zeigt, welche Daten in der Bewertung enthalten sind.

Weitere Informationen zur Verwendung CrowdStrike mit AWS Verified Access finden Sie auf der GitHub Website unter Absichern privater Anwendungen mit CrowdStrike und AWS Verified Access.

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

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die anhand der von CrowdStrike bereitgestellten Vertrauensdaten bewertet wird.

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

JumpCloud

JumpCloud ist ein vertrauenswürdiger Drittanbieter. Wenn Sie eine Richtlinie JumpCloud als Vertrauensanbieter definieren, schließt Verified Access die Vertrauensdaten im Cedar-Kontext unter dem Schlüssel ein, den Sie in der Trust-Provider-Konfiguration als „Richtlinien-Referenzname“ angeben. Wenn Sie möchten, können Sie eine Richtlinie schreiben, die anhand der Vertrauensdaten bewertet wird. Das folgende JSON-Schema zeigt, welche Daten in der Bewertung enthalten sind.

Weitere Informationen zur Verwendung JumpCloud mit AWS Verified Access finden Sie auf der JumpCloud Website unter Integrating JumpCloud and AWS Verified Access.

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

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die anhand des von bereitgestellten Vertrauenskontextes bewertet wird JumpCloud.

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

Der Benutzer behauptet, dass er weitergegeben und seine Unterschrift verifiziert hat

Nachdem eine AWS Verified Access-Instanz einen Benutzer erfolgreich authentifiziert hat, sendet sie die vom IdP empfangenen Benutzeransprüche an den Verified Access-Endpunkt. Die Benutzeransprüche werden signiert, sodass Anwendungen sowohl die Signaturen als auch überprüfen können, ob die Ansprüche von Verified Access gesendet wurden. Während dieses Vorgangs wird der folgende HTTP-Header hinzugefügt:

`x-amzn-ava-user-context`

Dieser Header enthält die Benutzeransprüche im Format JSON Web Token (JWT). Das JWT-Format umfasst einen Header, eine Nutzlast und eine Signatur (jeweils mit Base64-URL-Codierung). Verified Access verwendet ES384 (ECDSA-Signaturalgorithmus, der den SHA-384-Hash-Algorithmus verwendet), um die JWT-Signatur zu generieren.

Anwendungen können diese Angaben zur Personalisierung oder für andere benutzerspezifische Erlebnisse verwenden. Anwendungsentwickler sollten sich vor der Verwendung über den Grad der

Einzigartigkeit und Überprüfung der einzelnen Angaben durch den Identitätsanbieter informieren. Im Allgemeinen ist die `sub` Angabe der beste Weg, um einen bestimmten Benutzer zu identifizieren.

Inhalt

Beispiel: Signiertes JWT für OIDC-Benutzeransprüche

Die folgenden Beispiele zeigen, wie der Header und die Nutzlast für OIDC-Benutzeransprüche im JWT-Format aussehen werden.

Beispiel für einen 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)
}
```

Beispiel-Nutzlast:

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

Beispiel: Signiertes JWT für IAM Identity Center-Benutzeransprüche

Die folgenden Beispiele zeigen, wie der Header und die Nutzlast für IAM Identity Center-Benutzeransprüche im JWT-Format aussehen werden.

> **Note**
>
> Für IAM Identity Center werden nur Benutzerinformationen in den Ansprüchen enthalten sein.

Beispiel für einen 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)
}
```

Beispiel-Nutzlast:

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

Öffentliche Schlüssel

Da Verified Access-Instanzen Benutzeransprüche nicht verschlüsseln, empfehlen wir, Verified Access-Endpunkte für die Verwendung von HTTPS zu konfigurieren. Wenn Sie Ihren Verified Access-Endpunkt für die Verwendung von HTTP konfigurieren, achten Sie darauf, den Datenverkehr zum Endpunkt mithilfe von Sicherheitsgruppen zu beschränken.

Wir empfehlen Ihnen, die Signatur zu überprüfen, bevor Sie aufgrund der Behauptungen eine Autorisierung vornehmen. Sie erhalten den öffentlichen Schlüssel, indem Sie die Schlüssel-ID aus dem JWT-Header verwenden, um den öffentlichen Schlüssel aus dem Endpunkt zu suchen. Der Endpunkt für jeden AWS-Region lautet wie folgt:

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

Beispiel: JWT abrufen und dekodieren

Das folgende Codebeispiel zeigt, wie die Schlüssel-ID, der öffentliche Schlüssel und die Nutzlast in Python 3.9 abgerufen werden.

```
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'])
```

Verifizierte Zugriffsrichtlinien

AWSVerified Access-Richtlinien ermöglichen es Ihnen, Regeln für den Zugriff auf Ihre in gehosteten Anwendungen zu definierenAWS. Sie sind in Cedar, einer AWS Richtliniensprache, verfasst. Mit Cedar können Sie Richtlinien erstellen, die anhand des Vertrauenskontextes bewertet werden, der von den identitäts- oder gerätebasierten Vertrauensanbietern gesendet wird, die Sie für die Verwendung mit Verified Access konfiguriert haben.

Ausführlichere Informationen zur Sprache der Cedar-Richtlinien finden Sie im Cedar-Referenzhandbuch.

In diesem Abschnitt wird beschrieben, wie die Verified Access-Richtlinien strukturiert sind, was sie enthalten und wie sie zu definieren sind. Außerdem werden einige Beispiele aufgeführt.

Inhalt

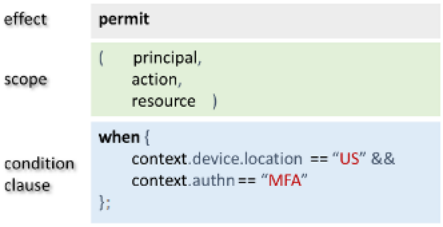
Arbeiten Sie mit Richtlinien für verifizierten Zugriff

Wenn Sie eine Verified Access-Gruppe oder einen Verified Access-Endpunkt erstellen, haben Sie die Möglichkeit, die Verified Access-Richtlinie zu definieren. Sie können eine Gruppe oder einen Endpunkt erstellen, ohne die Richtlinie für verifizierten Zugriff zu definieren. Alle Zugriffsanfragen werden jedoch blockiert, bis Sie eine Richtlinie definieren.

Informationen zum Hinzufügen oder Ändern einer Richtlinie für eine bestehende Gruppe oder einen Endpunkt mit verifiziertem Zugriff, nachdem dieser erstellt wurde, finden Sie unter Eine Gruppenverifizfizfizfizfizfizfizfizfiz oderÄndern Sie eine Endpunktrichtlinie für verifizierten Zugriff.

Struktur der Richtlinienerklärung

In diesem Abschnitt werden die Richtlinienerklärung „AWSVerified Access“ und ihre Bewertung beschrieben. In einer einzigen Verified Access-Richtlinie können mehrere Aussagen enthalten sein. Das folgende Diagramm zeigt die Struktur einer Verified Access-Richtlinie.



Die Richtlinie umfasst die folgenden Teile:

- Wirkung — Gibt an, ob die Richtlinienanweisung `permit` (`Allow`) oder `forbid` (`Deny`) lautet.
- Geltungsbereich — Gibt die Prinzipale, Aktionen und Ressourcen an, für die der Effekt gilt. Sie können den Geltungsbereich in Cedar undefiniert lassen, indem Sie bestimmte Prinzipale, Aktionen oder Ressourcen nicht identifizieren (wie im vorherigen Beispiel gezeigt). In diesem Fall gilt die Richtlinie für alle möglichen Prinzipale, Aktionen und Ressourcen.
- Bedingungsklausel — Gibt den Kontext an, in dem der Effekt gilt.

> **Important**
>
> Bei Verified Access werden Richtlinien vollständig ausgedrückt, indem in der Bedingungsklausel auf den Vertrauenskontext verwiesen wird. Der Geltungsbereich der Richtlinie muss immer undefiniert bleiben. In der Bedingungsklausel können Sie dann den Zugriff anhand der Identität und des Gerätevertrauenskontextes spezifizieren.

Einfaches Richtlinienbeispiel

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

Beachten Sie im vorherigen Beispiel, dass Sie mit dem && Operator mehr als eine Bedingungsklausel in einer Richtlinienanweisung verwenden können. Die Richtliniensprache von Cedar gibt Ihnen die Möglichkeit, maßgeschneiderte, detaillierte und umfangreiche Richtlinienerklärungen zu erstellen. Weitere Beispiele finden Sie unter Beispielrichtlinien.

Richtlinienevaluierung

Ein Strategiedokument besteht aus einer oder mehreren Grundsatzerklärungen (`permit`oder `forbid` Aussagen). Die Richtlinie gilt, wenn die Bedingungsklausel (die `when` Aussage) wahr ist. Damit ein Richtliniendokument den Zugriff ermöglicht, muss mindestens eine Genehmigungsrichtlinie in dem Dokument gelten, und es dürfen keine Verbotsrichtlinien gelten. Wenn keine Genehmigungsrichtlinien gelten und/oder eine oder mehrere Verbotsrichtlinien gelten, verweigert das Richtliniendokument den Zugriff. Wenn Sie Richtliniendokumente sowohl für die Verified Access-Gruppe als auch für den Verified Access-Endpunkt definiert haben, müssen beide Dokumente den Zugriff ermöglichen. Wenn Sie kein Richtliniendokument für den Verified Access-Endpunkt definiert haben, benötigt nur die Gruppenrichtlinie Verified Access Zugriff.

> **Note**
>
> AWSVerified Access validiert die Syntax, wenn Sie die Richtlinie erstellen, aber nicht die Daten, die Sie in die Bedingungsklausel eingeben.

Integrierte Operatoren

Wenn Sie den Kontext einer Richtlinie für AWS verifizierten Zugriff anhand verschiedener Bedingungen erstellen, wie unter beschriebenStruktur der Richtlinienerklärung, können Sie den && Operator verwenden, um zusätzliche Bedingungen hinzuzufügen. Es gibt auch viele andere integrierte Operatoren, mit denen Sie Ihren Versicherungsbedingungen zusätzliche Aussagekraft verleihen können. Die folgende Tabelle enthält alle integrierten Operatoren als Referenz.

Operator	Typen und Überladungen	Beschreibung
!	Boolean → Boolean	Logisch nicht.
==	beliebig → beliebig	Gleichheit. Funktioniert mit Argumenten aller Art,

Operator	Typen und Überladungen	Beschreibung
		auch wenn die Typen nicht übereinstimmen. Werte verschiedener Typen sind einander niemals gleich.
!=	beliebig → beliebig	Ungleichheit; das genaue Gegenteil von Gleichheit (siehe oben).
<	(lang, lang) → Boolesch	Lange Ganzzahl kleiner als.
<=	(lang, lang) → Boolesch	Lange Ganzzahl -to less-than-or-equal.
>	(lang, lang) → Boolean	Lange Ganzzahl größer als.
>=	(lang, lang) → Boolesch	Lange Ganzzahl -to greater-than-or-equal.
in	(Entität, Entität) → Boolean	Hierarchiezugehörigkeit (reflexiv: A in A ist immer wahr).
	(Entität, Menge (Entität)) → Boolean	Hierarchiezugehörigkeit: A in [B, C,...] ist wahr, wenn (A und B) || (A in C) ||... ein Fehler auftritt, wenn die Menge eine Nicht-Entität enthält.
&&	(Boolean, Boolean) → Boolean	Logisch und (kurzschließend).
||	(Boolean, Boolean) → Boolean	Logisch oder (Kurzschluss).
.existiert ()	Entität → Boolean	Existenz einer Entität.

Operator	Typen und Überladungen	Beschreibung
hat	(Entität, Attribut) → Boolean	Infix-Operator. `e has f` testet, ob der Datensatz oder die Entität eine Bindung für das Attribut `e` `f` hat. Gibt zurück`false`, ob es nicht `e` existiert oder ob `e` es existiert , aber das Attribut nicht hat`f`. Attribute können als Bezeichner oder Zeichenkettenliterale ausgedrückt werden.
like	(Zeichenfolge, Zeichenfolge) → Boolean	Infix-Operator. `t like p` prüft, ob der Text dem Muster `t` entsprichtp, das Platzhalterzeichen enthalten kann`*`, die 0 oder mehr eines beliebigen Zeichens entsprechen. Um einem buchstäblichen Sternzeichen in zu entsprechen`t`, können Sie die spezielle Escape-Zeichenfolge `\*` in verwenden. `p`
.enthält ()	(gesetzt, beliebig) → Boolean	Mitgliedschaft festlegen (ist B ein Element von A).
. enthält Alle ()	(set, set) → Boolean	Testet, ob Satz A alle Elemente in Satz B enthält.
. enthält Any ()	(Satz, Satz) → Boolean	Testet, ob Satz A eines der Elemente in Satz B enthält.

Kommentare zur Politik

Sie können Kommentare in Ihre Richtlinien für AWS verifizierten Zugriff aufnehmen. Kommentare sind als eine Zeile definiert, die mit einer neuen Zeile beginnt `//` und mit einer neuen Zeile endet.

Das folgende Beispiel zeigt Kommentaraussagen in der Richtlinie.

```
// 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)
};
```

Die Logik der Richtlinien wird kurzgeschlossen

Möglicherweise möchten Sie eine Richtlinie für AWS verifizierten Zugriff schreiben, die Daten auswertet, die in einem bestimmten Kontext möglicherweise vorhanden sind oder nicht. Wenn Sie Daten in einem Kontext referenzieren, der nicht existiert, erzeugt Cedar unabhängig von Ihrer Absicht einen Fehler und prüft die Richtlinie, um den Zugriff zu verweigern. Dies würde beispielsweise zu einer Ablehnung führen, da `fake_provider` und in diesem Kontext `bogus_key` nicht existieren.

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

Um diese Situation zu vermeiden, können Sie mithilfe des `has` Operators überprüfen, ob ein Schlüssel vorhanden ist. Wenn der `has` Operator False zurückgibt, wird die weitere Auswertung der verketteten Anweisung angehalten, und Cedar gibt keinen Fehler aus, wenn versucht wird, auf ein Element zu verweisen, das nicht existiert.

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

Dies ist besonders nützlich, wenn Sie eine Richtlinie angeben, die auf zwei verschiedene Vertrauensanbieter verweist.

```
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)
    )
  )
};
```

Beispielrichtlinien

Beispiel 1: Richtlinien für IAM Identity Center erstellen

> **Note**
>
> Da Gruppennamen geändert werden können, bezieht sich IAM Identity Center auf Gruppen, die ihre Gruppen-ID verwenden. Auf diese Weise wird vermieden, dass bei der Änderung des Gruppennamens gegen eine Richtlinienaussage verstoßen wird.

Die folgende Beispielrichtlinie ermöglicht den Zugriff nur, wenn ein Benutzer zu der `finance` Gruppe gehört (die die Gruppen-ID hat`c242c5b0-6081-1845-6fa8-6e0d9513c107`) und über eine verifizierte E-Mail-Adresse verfügt.

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

Beispiel 1b: Hinzufügen weiterer Bedingungen zu einer Richtlinienerklärung für IAM Identity Center

Die folgende Beispielrichtlinie erlaubt den Zugriff nur, wenn ein Benutzer der `finance` Gruppe angehört (die die Gruppen-ID hat`c242c5b0-6081-1845-6fa8-6e0d9513c107`), über eine verifizierte E-Mail-Adresse verfügt und die Risikobewertung für Geräte von Jamf lautet. `LOW`

```
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"
};
```

Beispiel 2: Dieselbe Richtlinie für einen OIDC-Drittanbieter

Die folgende Beispielrichtlinie erlaubt den Zugriff nur, wenn der Benutzer der Gruppe „Finanzen“ angehört, über eine verifizierte E-Mail-Adresse verfügt und die Risikobewertung für Geräte von Jamf NIEDRIG ist.

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

Beispiel 3: Verwenden CrowdStrike

Die folgende Beispielrichtlinie ermöglicht den Zugriff, wenn die Gesamtpunktzahl der Bewertung höher als 50 ist.

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

Beispiel 4: Arbeiten mit Sonderzeichen

Das folgende Beispiel zeigt, wie eine Richtlinie geschrieben wird, wenn eine Kontexteigenschaft ein `:` (Semikolon) verwendet, ein reserviertes Zeichen in der Richtliniensprache.

```
permit(principal, action, resource)
```

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

Beispiel 5: Eine bestimmte IP-Adresse zulassen

Das folgende Beispiel zeigt eine Richtlinie, die nur eine bestimmte IP-Adresse zulässt.

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

Beispiel 5a: Blockieren Sie eine bestimmte IP-Adresse

Das folgende Beispiel zeigt eine Richtlinie, die eine bestimmte IP-Adresse blockiert.

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

Assistent für verifizierte Zugriffsrichtlinien

Der Richtlinienassistent für verifizierten Zugriff ist ein Tool in der Verified Access-Konsole, mit dem Sie Ihre Richtlinien testen und entwickeln können. Er zeigt die Endpunktrichtlinie, die Gruppenrichtlinie und den Vertrauenskontext auf einem Bildschirm an, auf dem Sie die Richtlinien testen und bearbeiten können.

Die Formate des Vertrauenskontextes variieren je nach Vertrauensanbieter, und manchmal weiß der Administrator für verifizierten Zugriff möglicherweise nicht genau, welches Format ein bestimmter Vertrauensanbieter verwendet. Aus diesem Grund kann es für Test- und Entwicklungszwecke sehr hilfreich sein, den Vertrauenskontext und sowohl die Gruppen- als auch die Endpunktrichtlinien an einem Ort zu sehen.

In den folgenden Abschnitten werden die Grundlagen der Verwendung des Policy-Editors beschrieben.

Aufgaben

- Schritt 1: Geben Sie Ihre Ressourcen an

Schritt 1: Geben Sie Ihre Ressourcen an

Auf der ersten Seite des Richtlinienassistenten geben Sie den Verified Access-Endpunkt an, mit dem Sie arbeiten möchten. Sie geben auch einen Benutzer (identifiziert durch die E-Mail-Adresse) und optional den Namen des Benutzers und/oder eine Gerätekennung an. Standardmäßig wird die neueste Autorisierungsentscheidung aus den Verified Access-Protokollen für den angegebenen Benutzer extrahiert. Sie können optional die neueste Entscheidung zum Zulassen oder Verweigern speziell auswählen.

Schließlich werden der Vertrauenskontext, die Autorisierungsentscheidung, die Endpunktrichtlinie und die Gruppenrichtlinie auf dem nächsten Bildschirm angezeigt.

Um den Richtlinienassistenten zu öffnen und Ihre Ressourcen anzugeben

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access-Instances aus und klicken Sie dann auf die Verified Access-Instanz-ID für die Instanz, mit der Sie arbeiten möchten.
3. Wählen Sie Policy Assistant starten aus.
4. Geben Sie unter Benutzer-E-Mail-Adresse die E-Mail-Adresse des Benutzers ein.
5. Wählen Sie unter Verified Access-Endpunkt den Endpunkt aus, für den Sie Richtlinien bearbeiten und testen möchten.
6. (Optional) Geben Sie unter Name den Namen des Benutzers ein.
7. (Optional) Geben Sie unter Gerätekennung die eindeutige Gerätekennung ein.
8. (Optional) Wählen Sie unter Autorisierungsergebnis den Typ des letzten Autorisierungsergebnisses aus, das Sie verwenden möchten. Standardmäßig wird das neueste Autorisierungsergebnis verwendet.
9. Wählen Sie Weiter.

Schritt 2: Richtlinien testen und bearbeiten

Auf dieser Seite werden Ihnen die folgenden Informationen angezeigt, mit denen Sie arbeiten können:

- Der Vertrauenskontext, der von Ihrem Vertrauensanbieter für den Benutzer und (optional) das Gerät gesendet wurde, das Sie im vorherigen Schritt angegeben haben.
- Die Cedar-Richtlinie für den Verified Access-Endpunkt, die im vorherigen Schritt angegeben wurde.
- Die Cedar-Richtlinie für die Verified Access-Gruppe, zu der der Endpunkt gehört.

Die Cedar-Richtlinien für den Verified Access-Endpunkt und die Gruppe können auf dieser Seite bearbeitet werden, aber der Vertrauenskontext ist statisch. Sie können diese Seite jetzt verwenden, um den Vertrauenskontext zusammen mit den Cedar-Richtlinien anzuzeigen.

Testen Sie die Richtlinien anhand des Vertrauenskontextes, indem Sie auf die Schaltfläche Richtlinien testen klicken. Das Autorisierungsergebnis wird dann auf dem Bildschirm angezeigt. Sie können Änderungen an den Richtlinien vornehmen und Ihre Änderungen erneut testen und den Vorgang bei Bedarf wiederholen.

Wenn Sie mit den an den Richtlinien vorgenommenen Änderungen zufrieden sind, wählen Sie Weiter, um zum nächsten Bildschirm des Richtlinienassistenten zu gelangen.

Schritt 3: Überprüfen und übernehmen Sie die Änderungen

Auf der letzten Seite des Richtlinienassistenten werden die Änderungen, die Sie an den Richtlinien vorgenommen haben, zur leichteren Überprüfung hervorgehoben. Sie können sie nun ein letztes Mal überprüfen und auf Änderungen anwenden klicken, um die Änderungen zu übernehmen.

Sie haben auch die Möglichkeit, zur vorherigen Seite zurückzukehren, indem Sie Zurück wählen, oder den Richtlinienassistenten vollständig zu beenden, indem Sie Abbrechen wählen.

Sicherheit bei AWS verifiziertem Zugriff

Die Sicherheit in der Cloud hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und als Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für AWS Verified Access gelten, finden Sie unter AWSServices im Umfang nach Compliance-Programm AWS .
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Verified Access anwenden können. In den folgenden Themen erfahren Sie, wie Sie Verified Access konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Außerdem erfahren Sie, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer Verified Access-Ressourcen unterstützen.

Inhalt

Datenschutz in AWS Verified Access

Das Modell der AWS geteilten gilt für den Datenschutz in AWS Verified Access. https://aws.amazon.com/compliance/shared-responsibility-model/ Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS-Modell der geteilten Verantwortung und in der DSGVO im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Verified Access oder anderen AWS-Services über die Konsole, APIAWS CLI, oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung während der Übertragung

Verified Access verschlüsselt alle Daten während der Übertragung von Endbenutzern zu Verified Access-Endpunkten über das Internet mit Transport Layer Security (TLS) 1.2 oder höher.

Datenschutz für den Datenverkehr zwischen Netzwerken

Sie können Verified Access so konfigurieren, dass der Zugriff auf bestimmte Ressourcen in Ihrer VPC eingeschränkt wird. Für die benutzerbasierte Authentifizierung können Sie auch den Zugriff auf Teile Ihres Netzwerks einschränken, basierend auf der Benutzergruppe, die auf die Endpunkte zugreift. Weitere Informationen finden Sie unter Verifizierte Zugriffsrichtlinien.

Datenverschlüsselung im Ruhezustand für AWS Verified Access

AWS Verified Access verschlüsselt Daten im Ruhezustand standardmäßig mit AWS eigenen KMS-Schlüsseln. Wenn die Verschlüsselung von Daten im Ruhezustand standardmäßig erfolgt, trägt dies dazu bei, den betrieblichen Aufwand und die Komplexität zu reduzieren, die mit dem Schutz sensibler Daten verbunden sind. Gleichzeitig können Sie damit sichere Anwendungen erstellen, die strenge Verschlüsselungsvorschriften und gesetzliche Auflagen erfüllen. In den folgenden Abschnitten erfahren Sie, wie Verified Access KMS-Schlüssel für die Datenverschlüsselung im Ruhezustand verwendet.

Inhalt

Verifizierter Zugriff und KMS-Schlüssel

AWS-eigene Schlüssel

Verified Access verwendet KMS-Schlüssel, um persönlich identifizierbare Informationen (PII) automatisch zu verschlüsseln. Dies geschieht standardmäßig und Sie können die Verwendung der AWS-eigenen Schlüssel nicht selbst anzeigen, verwalten, verwenden oder überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme zum Schutz der Schlüssel ändern, die zur Verschlüsselung Ihrer Daten verwendet werden. Weitere Informationen finden Sie unter AWS-eigene Schlüssel im AWS Key Management Service-Entwicklerhandbuch.

Sie können diese Verschlüsselungsebene zwar nicht deaktivieren oder einen alternativen Verschlüsselungstyp auswählen, aber Sie können eine zweite Verschlüsselungsebene über den vorhandenen -AWSeigenen Verschlüsselungsschlüsseln hinzufügen, indem Sie beim Erstellen Ihrer Ressourcen für Verified Access einen vom Kunden verwalteten Schlüssel auswählen.

Kundenverwaltete Schlüssel

Verified Access unterstützt die Verwendung von symmetrischen, vom Kunden verwalteten Schlüsseln, die Sie erstellen und verwalten, um eine zweite Verschlüsselungsebene gegenüber der vorhandenen Standardverschlüsselung hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:

- Festlegung und Pflege wichtiger Richtlinien
- Festlegung und Aufrechterhaltung von IAM-Richtlinien und -Zuschüssen
- Aktivieren und Deaktivieren wichtiger Richtlinien
- Kryptographisches Material mit rotierendem Schlüssel
- Hinzufügen von Tags
- Erstellen von Schlüsselaliasen
- Schlüssel für das Löschen von Schlüsseln planen

Weitere Informationen finden Sie unter Kundenverwaltete Schlüssel im AWS Key Management ServiceEntwicklerhandbuch.

> **Note**
>
> Verified Access aktiviert automatisch die Verschlüsselung im Ruhezustand mit AWS-eigenen Schlüsseln, um persönlich identifizierbare Daten kostenlos zu schützen.

Es fallen jedoch AWS KMS Gebühren an, wenn Sie einen vom Kunden verwalteten Schlüssel verwenden. Weitere Informationen zu Preisen finden Sie unter AWS Key Management Service – Preise.

Persönlich identifizierbare Informationen

Die folgende Tabelle fasst die persönlich identifizierbaren Informationen (PII) zusammen, die Verified Access verwendet, und wie sie verschlüsselt werden.

Datentyp	AWS Verschlüsselung von -eigenen Schlüsseln	Vom Kunden verwaltete Schlüsselverschlüsselung (optional)
`Trust provider (user-type)` Vertrauensanbieter vom Benutzertyp enthalten OIDC-Optionen wie AuthorizationEndpoint, UserInfoEndpoint ClientId, ClientSecret, usw., die als PII betrachtet werden.	Aktiviert	Aktiviert
`Trust provider (device-type)` Vertrauensanbieter vom Gerätetyp enthalten eine TenantId, die als PII betrachtet wird.	Aktiviert	Aktiviert
`Group policy` Wird bei der Erstellung oder Änderung der Gruppe mit verifiziertem Zugriff bereitgestellt. Enthält Regeln zum	Aktiviert	Aktiviert

Datentyp	AWS Verschlüsselung von - eigenen Schlüsseln	Vom Kunden verwaltete Schlüsselverschlüsselung (optional)
Autorisieren von Zugriffsa nforderungen. Kann PII wie Benutzername und E-Mail-Ad resse usw. enthalten.		
`Endpoint policy` Wird bei der Erstellung oder Änderung des Endpunkts für verifizierten Zugriff bereitges tellt. Enthält Regeln zum Autorisieren von Zugriffsa nforderungen. Kann PII wie Benutzername und E-Mail-Ad resse usw. enthalten.	Aktiviert	Aktiviert

So verwendet AWS Verified Access Erteilungen in AWS KMS

Verified Access erfordert eine Erteilung, um Ihren vom Kunden verwalteten Schlüssel zu verwenden.

Wenn Sie Ressourcen für verifizierten Zugriff erstellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt sind, erstellt Verified Access in Ihrem Namen eine Erteilung, indem es eine CreateGrant Anfrage an sendetAWS KMS. Erteilungen in AWS KMS werden verwendet, um Verified Access Zugriff auf einen vom Kunden verwalteten Schlüssel in Ihrem Konto zu gewähren.

Verified Access erfordert die Erteilung, um Ihren vom Kunden verwalteten Schlüssel für die folgenden internen Vorgänge zu verwenden:

- Senden Sie Entschlüsselungsanfragen an AWS KMS, um die verschlüsselten Datenschlüssel zu entschlüsseln, damit sie zum Entschlüsseln Ihrer Daten verwendet werden können.
- Senden Sie RetireGrant Anfragen an AWS KMS, um eine Erteilung zu löschen.

Sie können den Zugriff auf die Genehmigung jederzeit widerrufen oder den Zugriff des Services auf den vom Kunden verwalteten Schlüssel entfernen. Wenn Sie dies tun, kann Verified Access nicht auf

die Daten zugreifen, die mit dem vom Kunden verwalteten Schlüssel verschlüsselt sind, was sich auf Operationen auswirkt, die von diesen Daten abhängig sind.

Verwenden von kundenverwalteten Schlüsseln mit Verified Access

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel erstellen, indem Sie die AWS Management Console oder die AWS KMS-APIs verwenden. Folgen Sie den Schritten zum Erstellen eines symmetrischen kundenverwalteten Schlüssels im Entwicklerhandbuch zum AWS Key Management Service.

Schlüsselrichtlinien

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter Verwalten des Zugriffs auf kundenverwaltete Schlüssel im Entwicklerhandbuch zum AWS Key Management Service.

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren Ressourcen für verifizierten Zugriff zu verwenden, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zulässig sein:

- `kms:CreateGrant`: Fügt einem kundenverwalteten Schlüssel eine Erteilung hinzu. Gewährt Kontrollzugriff auf einen angegebenen KMS-Schlüssel, der den Zugriff auf Erteilungsvorgänge ermöglicht, die Verified Access benötigt. Weitere Informationen zur Verwendung von Grants finden Sie im AWS Key Management Service -Entwicklerhandbuch.

  Auf diese Weise kann Verified Access Folgendes tun:

  - `GenerateDataKeyWithoutPlainText` aufrufen, um einen verschlüsselten Datenschlüssel zu generieren und zu speichern, da der Datenschlüssel nicht sofort zum Verschlüsseln verwendet wird.
  - `Decrypt` aufrufen, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf verschlüsselte Daten zu verwenden.
  - Richten Sie einen ausscheidenden Prinzipal ein, um dem Service zu erlaubenRetireGrant.
- `kms:DescribeKey` – Stellt die vom Kunden verwalteten Schlüsseldetails bereit, damit Verified Access den Schlüssel validieren kann.
- `kms:GenerateDataKey` – Ermöglicht Verified Access die Verwendung des Schlüssels zum Verschlüsseln von Daten.

- `kms:Decrypt` – Erlauben Sie Verified Access, die verschlüsselten Datenschlüssel zu entschlüsseln.

Im Folgenden finden Sie ein Beispiel für eine Schlüsselrichtlinie, die Sie für Verified Access verwenden können.

```
"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" : "*"
    }
  ]
```

Weitere Informationen zum Festlegen von Berechtigungen in einer Richtlinie finden Sie im AWS Key Management Service-Entwicklerhandbuch.

Weitere Informationen zur Fehlerbehebung beim Schlüsselzugriff finden Sie im AWS Key Management Service-Entwicklerhandbuch.

Angeben eines vom Kunden verwalteten Schlüssels für Ressourcen mit verifiziertem Zugriff

Sie können einen vom Kunden verwalteten Schlüssel angeben, um eine zweite Verschlüsselungsebene für die folgenden Ressourcen bereitzustellen:

- Verifizierte Zugriffsgruppe
- Verified Access-Endpunkt
- Vertrauensanbieter für verifizierten Zugriff

Wenn Sie eine dieser Ressourcen mit der erstellenAWS Management Console, können Sie im Abschnitt Zusätzliche Verschlüsselung – optional einen vom Kunden verwalteten Schlüssel angeben. Aktivieren Sie während des Vorgangs das Kontrollkästchen Verschlüsselungseinstellungen anpassen (erweitert) und geben Sie dann die AWS KMS Schlüssel-ID ein, die Sie verwenden möchten. Dies kann auch beim Ändern einer vorhandenen Ressource oder mithilfe der erfolgenAWS CLI.

> **Note**
>
> Wenn der vom Kunden verwaltete Schlüssel, der zum Hinzufügen zusätzlicher Verschlüsselung zu einer der oben genannten Ressourcen verwendet wird, verloren geht, sind die Konfigurationswerte für die Ressourcen nicht mehr zugänglich. Die Ressourcen können jedoch mithilfe der AWS Management Console oder geändert werden, um einen

neuen AWS CLIvom Kunden verwalteten Schlüssel anzuwenden und die Konfigurationswerte zurückzusetzen.

AWS Verschlüsselungskontext für verifizierten Zugriff

Ein Verschlüsselungskontext ist ein optionaler Satz von Schlüssel-Wert-Paaren, der zusätzliche Kontextinformationen zu den Daten enthalten kann. AWS KMS verwendet den Verschlüsselungskontext als Additional Authenticated Data (AAD) zur Unterstützung der authentifizierten Verschlüsselung. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, bindet AWS KMS den Verschlüsselungskontext an die verschlüsselten Daten. Zur Entschlüsselung von Daten müssen Sie denselben Verschlüsselungskontext in der Anfrage übergeben.

AWS Verschlüsselungskontext für verifizierten Zugriff

Verified Access verwendet in allen kryptografischen Operationen denselben AWS KMS Verschlüsselungskontext, wobei der Schlüssel `aws:verified-access:arn` und der Wert der Amazon-Ressourcenname (ARN) der Ressource ist. Im Folgenden finden Sie die Verschlüsselungskontexte für Ressourcen mit verifiziertem Zugriff.

Vertrauensanbieter für verifizierten Zugriff

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

Verifizierte Zugriffsgruppe

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

Verified Access-Endpunkt

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

```
}
```

Weitere Informationen zur Verwendung des Verschlüsselungskontexts für Erteilungen oder in Richtlinien finden Sie unter Verschlüsselungskontext im AWS Key Management Service Entwicklerhandbuch für .

Überwachen Ihrer Verschlüsselungsschlüssel für AWS Verified Access

Wenn Sie einen vom Kunden verwalteten KMS-Schlüssel mit Ihren AWS Ressourcen für verifizierten Zugriff verwenden, können Sie verwenden, AWS CloudTrail um Anforderungen zu verfolgen, die Verified Access an sendetAWS KMS.

Die folgenden Beispiele sind AWS CloudTrail Ereignisse für `CreateGrant`, `RetireGrant`, `DescribeKey`, und `GenerateDataKey`, die KMS`Decrypt`-Operationen überwachen, die von Verified Access aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten KMS-Schlüssel verschlüsselt sind:

CreateGrant

Wenn Sie einen vom Kunden verwalteten Schlüssel zum Verschlüsseln Ihrer Ressourcen verwenden, sendet Verified Access in Ihrem Namen eine `CreateGrant` Anforderung für den Zugriff auf den Schlüssel in Ihrem AWS Konto. Die Erteilung, die Verified Access erstellt, ist spezifisch für die Ressource, die dem vom Kunden verwalteten Schlüssel zugeordnet ist.

Das folgende Beispielereignis zeichnet den Vorgang `CreateGrant` auf:

```
{
    "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

Verified Access verwendet die -`RetireGrant`Operation, um eine Erteilung zu entfernen, wenn Sie eine Ressource löschen.

Das folgende Beispielereignis zeichnet den Vorgang `RetireGrant` auf:

```
{
    "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

Verified Access ruft die -`Decrypt`Operation auf, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf die verschlüsselten Daten zu verwenden.

Das folgende Beispielereignis zeichnet den Vorgang `Decrypt` auf:


```
{
    "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

Verified Access verwendet die -`DescribeKey`Operation, um zu überprüfen, ob der kundenverwaltete Schlüssel, der Ihrer Ressource zugeordnet ist, im Konto und in der Region vorhanden ist.

Das folgende Beispielereignis zeichnet den Vorgang `DescribeKey` auf:

```
{
    "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

Das folgende Beispielereignis zeichnet den Vorgang `GenerateDataKey` auf:

```
{
    "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"
}
```

Identitäts- und Zugriffsmanagement für AWS Verified Access

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Verified Access-Ressourcen zu verwenden. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Verified Access ausführen.

Dienstbenutzer — Wenn Sie den Verified Access-Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr Funktionen von Verified Access verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Featuresweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie unter Verifizierter Zugriff nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unterFehlerbehebung bei AWS verifizierter Zugriffsidentität und Zugriff.

Dienstadministrator — Wenn Sie in Ihrem Unternehmen für die Ressourcen mit verifiziertem Zugriff verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Verified Access. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen mit verifiziertem Zugriff Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um

die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit Verified Access verwenden kann, finden Sie unterSo funktioniert AWS Verified Access mit IAM.

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Verified Access schreiben können. Beispiele für identitätsbasierte Richtlinien für verifizierten Zugriff, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access AWS

Authentifizierung mit Identitäten

Authentifizierung ist die Art, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center. (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffsportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuerten Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anforderungen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter Signieren von AWS-API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center-Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode empfiehlt es sich, menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, aufzufordern, den Verbund mit einem Identitätsanbieter zu verwenden, um auf AWS-Services mit temporären Anmeldeinformationen zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus dem Benutzerverzeichnis Ihres Unternehmens, ein Web Identity Provider, AWS Directory Service, das Identity-Center-Verzeichnis oder jeder Benutzer, der mit Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt werden, auf AWS-Services zugreift. Wenn Verbundidentitäten auf AWS-Konten zugreifen, übernehmen sie Rollen und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen im IAM Identity Center erstellen oder Sie können eine Verbindung mit einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie in allen AWS-Konten und Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges

Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie Rollen wechseln. Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center-Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto

zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff – Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in Amazon Managed Service for Prometheus verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anforderungen werden nur dann gestellt, wenn ein Dienst eine Anforderung erhält, die Interaktionen mit anderen AWS-Services oder Ressourcen erfordert, um abgeschlossen werden zu können. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Serviceverknüpfte Rolle – Eine serviceverknüpfte Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.
- Anwendungen in Amazon EC2 – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die

Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern,

welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Dienste, die ACLs unterstützen. Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Ein ausdrückliches Ablehnen in einer dieser Richtlinien setzt das Zulassen außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Dienst für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organisationen und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations-Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter Logik für die Richtlinienauswertung im IAM-Benutzerhandbuch.

So funktioniert AWS Verified Access mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf Verified Access verwenden, sollten Sie sich darüber informieren, welche IAM-Funktionen für Verified Access verfügbar sind.

IAM-Funktionen, die Sie mit Verified Access verwenden können AWS

IAM-Funktion	Unterstützung für verifizierten Zugriff
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie Verified Access und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für verifizierten Zugriff

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für verifizierten Zugriff

Beispiele für identitätsbasierte Richtlinien für verifizierten Zugriff finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access AWS

Ressourcenbasierte Richtlinien innerhalb von Verified Access

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource in unterschiedlichen AWS-Konten befinden, muss ein IAM-Administrator im vertrauenswürdigen Konto auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource

erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Richtlinienmaßnahmen für Verified Access

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keine passende API-Operation gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Aktionen mit verifiziertem Zugriff finden Sie unter Von Amazon EC2 definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in Verified Access verwenden vor der Aktion das folgende Präfix:

```
ec2
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Beispiele für identitätsbasierte Richtlinien für verifizierten Zugriff finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access AWS

Richtlinienressourcen für verifizierten Zugriff

Unterstützt Richtlinienressourcen	Ja

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der Ressourcentypen mit verifiziertem Zugriff und ihren ARNs finden Sie unter Von Amazon EC2 definierte Ressourcen in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon EC2 definierte Aktionen.

Beispiele für identitätsbasierte Richtlinien für verifizierten Zugriff finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access AWS

Bedingungsschlüssel für Richtlinien für verifizierten Zugriff

Unterstützt servicespezifische Richtlini enbedingungsschlüssel	Ja

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet AWS die Bedingung mittels einer logischen `OR`-Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Eine Liste aller globalen AWS-Bedingungsschlüssel finden Sie unter Globale AWS-Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Eine Liste der Bedingungsschlüssel für verifizierten Zugriff finden Sie unter Bedingungsschlüssel für Amazon EC2 in der Service Authorization Reference. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon EC2 definierte Aktionen.

Beispiele für identitätsbasierte Richtlinien für verifizierten Zugriff finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access AWS

ACLs im Bereich Verifizierter Zugriff

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit verifiziertem Zugriff

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und mehrere AWS-Ressourcen anfügen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder `aws:TagKeys`Bedingung verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen mit verifiziertem Zugriff verwenden

Unterstützt temporäre Anmeldeinformationen	Ja

Einige AWS-Services Featureieren nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, darunter welche AWS-Services mit temporären Anmeldeinformationen Featureieren, finden Sie unter AWS-Services, die mit IAM Featureieren im IAM-Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen Methode als einem Benutzernamen und einem Passwort bei der AWS Management Console anmelden. Wenn

Sie beispielsweise über den Single Sign-On (SSO)-Link Ihres Unternehmens auf AWS zugreifen, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Sie können mithilfe der AWS CLI- oder AWS-API manuell temporäre Anmeldeinformationen erstellen. Sie können dann diese temporären Anmeldeinformationen verwenden, um auf AWS zuzugreifen. AWS empfiehlt, dass Sie temporäre Anmeldeinformationen dynamisch generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Dienstübergreifende Prinzipalberechtigungen für verifizierten Zugriff

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anforderungen werden nur dann gestellt, wenn ein Dienst eine Anforderung erhält, die Interaktionen mit anderen AWS-Services oder Ressourcen erfordert, um abgeschlossen werden zu können. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für verifizierten Zugriff

Unterstützt Servicerollen	Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

Mit Diensten verknüpfte Rollen für verifizierten Zugriff

Unterstützt serviceverknüpfte Rollen	Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen mit verifiziertem Zugriff finden Sie unter. Verwenden Sie dienstverknüpfte Rollen für verifizierten Zugriff

Beispiele für identitätsbasierte Richtlinien für Verified Access AWS

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Ressourcen mit verifiziertem Zugriff zu erstellen oder zu ändern. Sie können auch keine Aufgaben über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu den von Verified Access definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter Aktionen, Ressourcen und Zustandsschlüssel für Amazon EC2 in der Service Authorization Reference.

Themen

- Bewährte Methoden für Richtlinien
- Richtlinie für die Erstellung von Instanzen mit verifiziertem Zugriff
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Ressourcen mit verifiziertem Zugriff in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS-verwaltete Richtlinien oder AWS-verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.

- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Richtlinie für die Erstellung von Instanzen mit verifiziertem Zugriff

Um eine Verified Access-Instanz zu erstellen, müssen IAM-Prinzipale diese zusätzliche Erklärung zu ihrer IAM-Richtlinie hinzufügen.

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

> **Note**
>
> `verified-access:AllowVerifiedAccess`ist eine virtuelle API, die nur für Aktionen verwendet werden kann. Sie unterstützt keine auf Ressourcen-, Tag- oder Bedingungsschlüsseln basierende Autorisierung. Verwenden Sie für die API-Aktion eine auf Ressourcen-, Tag- oder Bedingungsschlüsseln basierende Autorisierung. `ec2:CreateVerifiedAccessInstance`

Beispielrichtlinie für die Erstellung einer Verified Access-Instanz. In diesem Beispiel `123456789012` handelt es sich um die AWS Kontonummer und `us-east-1` die AWS Region.

```
{
      "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": "*"
        }
    ]
}
```

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
    "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": "*"
        }
    ]
}
```

Fehlerbehebung bei AWS verifizierter Zugriffsidentität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Verified Access und IAM auftreten können.

Problembereiche

- Ich bin nicht berechtigt, eine Aktion in Verified Access durchzuführen
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Verified Access-Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion in Verified Access durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `ec2:`*`GetWidget`*-Berechtigungen verfügt.

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

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `ec2:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Durchführung der `iam:PassRole` Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Verified Access übergeben können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Dienst, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen Verified Access `marymajor` versucht, über die Konsole eine Aktion auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Verified Access-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Verified Access diese Funktionen unterstützt, finden Sie unterSo funktioniert AWS Verified Access mit IAM.
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto im IAM-Benutzerhandbuch.

- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter Gewähren des Zugriffs auf AWS-Konten von externen Benutzern im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Verwenden Sie dienstverknüpfte Rollen für verifizierten Zugriff

AWSVerified Access verwendet dienstverknüpfte AWS Identity and Access Management (IAM) Rollen. Eine dienstverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Verified Access verknüpft ist. Dienstverknüpfte Rollen sind von Verified Access vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere in AWS-Services Ihrem Namen anzurufen.

Eine dienstverknüpfte Rolle erleichtert die Einrichtung von Verified Access, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Verified Access definiert die Berechtigungen seiner dienstverknüpften Rollen, und sofern nicht anders definiert, kann nur Verified Access seine Rollen übernehmen. Zu den definierten Berechtigungen gehören die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen IAM-Entität zugeordnet werden.

Informationen zu anderen Services, die serviceorientierte Rollen unterstützen, finden Sie unter AWS services that work with IAM (-Services, die mit IAM funktionieren). Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Service-linked roles (Serviceorientierte Rollen) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Verified Access

Verified Access verwendet die mit dem Dienst verknüpfte Rolle AWSServiceRoleForVPCVerifiedAccess, um Ressourcen in Ihrem Konto bereitzustellen, die für die Nutzung des Dienstes erforderlich sind.

Die serviceverknüpfte Rolle AWSServiceRoleForVPCVerifiedAccess vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `verified-access.amazonaws.com`

Die benannte Richtlinie für Rollenberechtigungen ermöglicht es Verified Access AWSVPCVerifiedAccessServiceRolePolicy, die folgenden Aktionen für die angegebenen Ressourcen auszuführen:

- Aktion `ec2:CreateNetworkInterface` in allen Subnetzen und Sicherheitsgruppen sowie allen Netzwerkschnittstellen mit dem Tag `VerifiedAccessManaged=true`
- Aktion `ec2:CreateTags` auf allen Netzwerkschnittstellen zum Zeitpunkt der Erstellung
- Aktion `ec2:DeleteNetworkInterface` auf allen Netzwerkschnittstellen mit dem Tag `VerifiedAccessManaged=true`
- Aktion `ec2:ModifyNetworkInterfaceAttribute` für alle Sicherheitsgruppen und alle Netzwerkschnittstellen mit dem Tag `VerifiedAccessManaged=true`

Sie können die Berechtigungen für diese Richtlinie auch im AWS Management Console AWSVPCVerifiedAccessServiceRolePolicyoder im AWSManaged AWSVPCVerifiedAccessServiceRolePolicyPolicy Reference Guide einsehen.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Eine serviceverknüpfte Rolle für Verified Access erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie dieAWS Management Console, die oder die AWS API aufrufen CreateVerifiedAccessEndpointAWS CLI, erstellt Verified Access die mit dem Service verknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie CreateVerifiedAccessEndpointerneut anrufen, erstellt Verified Access die dienstverknüpfte Rolle erneut für Sie.

Eine dienstverknüpfte Rolle für Verified Access bearbeiten

Bei Verified Access können Sie die mit dem AWSServiceRoleForVPCVerifiedAccessDienst verknüpfte Rolle nicht bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert

werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen Sie eine dienstverknüpfte Rolle für Verified Access

Sie müssen die Rolle AWSServiceRoleForVPCVerifiedAccess nicht manuell löschen. Wenn Sie dieAWS Management Console, die oder die AWS API aufrufen DeleteVerifiedAccessEndpointAWS CLI, bereinigt Verified Access die Ressourcen und löscht die mit dem Service verknüpfte Rolle für Sie.

So löschen Sie die servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die AWSServiceRoleForVPCVerifiedAccess serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für mit dem Verified Access-Service verknüpfte Rollen

Verified Access unterstützt die Verwendung von dienstverknüpften Rollen überall AWS-Regionen dort, wo der Dienst verfügbar ist. Weitere Informationen finden Sie unter AWSRegionen und Endpunkte.

AWSverwaltete Richtlinien für AWS verifizierten Zugriff

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AWSVPCVerifiedAccessServiceRolePolicy

Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es Verified Access ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen verwenden. Die Berechtigungen für diese Richtlinie finden Sie AWSVPCVerifiedAccessServiceRolePolicyim oder Sie können die AWS Management Console AWSVPCVerifiedAccessServiceRolePolicyRichtlinie im Referenzhandbuch für AWS verwaltete Richtlinien einsehen.

Updates für AWS verwaltete Richtlinien mit verifiziertem Zugriff

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien für Verified Access, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen. Wenn Sie automatische Benachrichtigungen über Änderungen an dieser Seite erhalten möchten, abonnieren Sie den RSS-Feed auf der Seite mit dem Verlauf der Verified Access-Dokumente.

Änderung	Beschreibung	Datum
AWSVPCVerifiedAccessServiceRolePolicy- Die Richtlinie wurde aktualisiert	Verified Access hat die verwaltete Richtlinie aktualisiert und enthält nun Beschreibungen aller Aktionen im Feld „Sid“.	17. November 2023
AWSVPCVerifiedAccessServiceRolePolicy- Die Richtlinie wurde aktualisiert	Verified Access hat seine verwaltete Richtlinie aktualisiert, um der `ec2:CreateNetworkInterface` Berechtigung eine Sicherheitsgruppenressource hinzuzufügen.	31. Mai 2023
AWSVPCVerifiedAccessServiceRolePolicy – Neue Richtlinie	Verified Access hat eine neue Richtlinie hinzugefügt, die es ermöglicht, Ressourcen in Ihrem Konto bereitzus	29. November 2022

Änderung	Beschreibung	Datum
	tellen, die für die Nutzung des Dienstes erforderlich sind.	
Verified Access hat mit der Nachverfolgung von Änderungen begonnen	Verified Access hat damit begonnen, Änderungen an den AWS verwalteten Richtlini en nachzuverfolgen.	29. November 2022

Konformitätsprüfung für AWS verifizierten Zugriff

AWS Verified Accesskann so konfiguriert werden, dass die Einhaltung der Federal Information Processing Standards (FIPS) unterstützt wird. Weitere Informationen und Einzelheiten zur Einrichtung der FIPS-Konformität für Verified Access finden Sie unter. FIPS-Konformität für verifizierten Zugriff

Informationen darüber, ob ein AWS-Service in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter AWS-Services in Geltungsbereich nach Compliance-Programm. Wählen Sie das Compliance-Programm, das Sie interessiert. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Compliance-Verantwortung bei der Verwendung von AWS-Services ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- Kurzanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden werden Überlegungen zur Architektur erörtert und Schritte zum Bereitstellen von Basisumgebungen auf AWS zur Verfügung gestellt, die auf Sicherheit und Compliance ausgerichtet sind.
- Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance in Amazon Web Services – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-berechtigte Anwendungen erstellen können.

> **Note**
>
> Nicht alle AWS-Services sind HIPAA-berechtigt. Weitere Informationen finden Sie in der Referenz für HIPAA-berechtigte Services.

- AWS-Compliance-Ressourcen – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- AWS-Compliance-Leitfäden für Kunden – Verstehen Sie das Modell der geteilten Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Methoden zum Schutz von AWS-Services zusammengefasst und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Auswertung von Ressourcen mit Regeln im AWS ConfigEntwicklerhandbuch – Der AWS Config-Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.
- AWS Security Hub – Dieser AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb von AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS-Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- AWS Audit Manager – Dieser AWS-Service hilft Ihnen, Ihre AWS-Nutzung kontinuierlich zu überprüfen, um den Umgang mit Risiken und die Compliance von Branchenstandards zu vereinfachen.

Resilienz bei AWS verifiziertem Zugriff

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter AWSGlobale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur bietet Verified Access die folgende Funktion, um Ihre Hochverfügbarkeitsanforderungen zu erfüllen.

Mehrere Subnetze für hohe Verfügbarkeit

Wenn Sie einen Verified Access-Endpunkt vom Typ Load Balancer erstellen, können Sie dem Endpunkt mehrere Subnetze zuordnen. Jedes Subnetz, das Sie dem Endpunkt zuordnen, muss zu einer anderen Availability Zone gehören. Durch die Zuordnung mehrerer Subnetze können Sie eine hohe Verfügbarkeit sicherstellen, indem Sie mehrere Availability Zones verwenden.

Überwachung des AWS verifizierten Zugriffs

Die Überwachung ist wichtig, um die Zuverlässigkeit, Verfügbarkeit und Leistung von AWS VeriWatch aufrechtzuerhalten. AWSstellt die folgenden Überwachungstools bereit, um den verifizierten Zugriff zu überwachen, Sie zu informieren, wenn etwas nicht stimmt, und um gegebenenfalls automatische Aktionen durchzuführen:

- Zugriffsprotokolle — Erfassen Sie detaillierte Informationen zu Anfragen für den Zugriff auf Anwendungen. Weitere Informationen finden Sie unter the section called “Protokolle für verifizierten Zugriff”.
- AWS CloudTrail— Erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen von Ihnen erfolgten, AWS-Konto und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie unter the section called “CloudTrail-Protokolle”.

Protokolle für verifizierten Zugriff

Nachdem AWS Verified Access jede Zugriffsanforderung ausgewertet hat, protokolliert es alle Zugriffsversuche. Dies bietet einen zentralen Einblick in den Anwendungszugriff und hilft Ihnen, schnell auf Sicherheitsvorfälle und Prüfungsanforderungen zu reagieren. Verified Access unterstützt das Open Cybersecurity Schema Framework (O microSD)-Protokollierungsformat.

Wenn Sie die Protokollierung aktivieren, müssen Sie ein Ziel für das Senden der Protokolle konfigurieren. Der IAM-Prinzipal, der zur Konfiguration des Protokollierungsziels verwendet wird, muss über bestimmte Berechtigungen verfügen, damit die Protokollierung ordnungsgemäß funktioniert. Die erforderlichen IAM-Berechtigungen für jedes Protokollierungsziel finden Sie im Protokollierungsberechtigungen Abschnitt . Verified Access unterstützt die folgenden Ziele für die Veröffentlichung von Zugriffsprotokollen:

- Amazon- CloudWatch Logs-Protokollgruppen
- Amazon-S3-Buckets
- Bereitstellungsdatenströme von Amazon Data Firehose

Inhalt

Protokollierungsversionen

Standardmäßig verwendet das Protokollierungssystem für verifizierten Zugriff Open Cybersecurity Schema Framework (O microSD) Version 0.1. Beispielprotokolle mit Version 0.1 finden Sie im Beispiele für O microSD Version 0.1 Abschnitt .

Die neueste Protokollierungsversion ist mit O microSD Version 1.0.0-rc.2 kompatibel. Spezifische Details zur im Schema finden Sie hier O microSD Schema . Beispielprotokolle mit Version 1.0.0-rc.2 finden Sie im Beispiele für O microSD Version 1.0.0-rc.2 Abschnitt .

Upgrade-Protokollierungsversion

Wenn Sie die verwendete Protokollierungsversion aktualisieren möchten, gehen Sie wie folgt vor.

So aktualisieren Sie die Protokollierungsversion mithilfe der Konsole

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Instances mit verifiziertem Zugriff aus.
3. Wählen Sie die entsprechende Instance mit verifiziertem Zugriff aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff die Option Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff ändern aus.
5. Wählen Sie ocsf-1.0.0-rc.2 aus der Dropdown-Liste Protokollversion aktualisieren aus.
6. Wählen Sie Konfiguration für die Instance-Protokollierung mit verifiziertem Zugriff ändern aus.

So aktualisieren Sie die Protokollierungsversion mithilfe der AWS CLI

Verwenden Sie den Befehl modify-verified-access-instance-logging-configuration.

Protokollierungsberechtigungen

Der IAM-Prinzipal, der zur Konfiguration des Protokollierungsziels verwendet wird, muss über bestimmte Berechtigungen verfügen, damit die Protokollierung ordnungsgemäß funktioniert. Im Folgenden finden Sie die Berechtigungen, die für jedes Protokollierungsziel erforderlich sind.

Für die Übermittlung an - CloudWatch Protokolle:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration` auf der Instance mit verifiziertem Zugriff
- `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDeliverylogs:ListLogDeliveries`, und `logs:UpdateLogDelivery` auf allen Ressourcen
- `logs:DescribeLogGroups`, `logs:DescribeResourcePoliciesund` `logs:PutResourcePolicy` in der Zielprotokollgruppe

Für die Übermittlung an Amazon S3:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration` auf der Instance mit verifiziertem Zugriff
- `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDeliverylogs:ListLogDeliveries`, und `logs:UpdateLogDelivery` auf allen Ressourcen
- `s3:GetBucketPolicy` und `s3:PutBucketPolicy` im Ziel-Bucket

Für die Bereitstellung an Firehose:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration` auf der Instance mit verifiziertem Zugriff
- `firehose:TagDeliveryStream` für alle Ressourcen
- `iam:CreateServiceLinkedRole` für alle Ressourcen
- `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDeliverylogs:ListLogDeliveries`, und `logs:UpdateLogDelivery` auf allen Ressourcen

Aktivieren oder Deaktivieren von Protokollen

Wenn Sie die Protokollierung aktivieren, müssen Sie ein Ziel für das Senden der Protokolle konfigurieren. Der IAM-Prinzipal, der zur Konfiguration des Protokollierungsziels verwendet wird, muss über bestimmte Berechtigungen verfügen, damit die Protokollierung ordnungsgemäß funktioniert. Die erforderlichen IAM-Berechtigungen für jedes Protokollierungsziel finden Sie im Protokollierungsberechtigungen Abschnitt .

Inhalt

Aktivieren der Zugriffsprotokolle

So aktivieren Sie Protokolle mit verifiziertem Zugriff

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Instances mit verifiziertem Zugriff aus.
3. Wählen Sie die Instance Verified Access aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff die Option Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff ändern aus.
5. (Optional) Gehen Sie wie folgt vor, um von Vertrauensanbietern gesendete Vertrauensdaten in die Protokolle aufzunehmen:
   a. Wählen Sie ocsf-1.0.0-rc.2 aus der Dropdown-Liste Protokollversion aktualisieren aus.
   b. Wählen Sie Vertrauenskontext einschließen aus.
6. Führen Sie eine der folgenden Aktionen aus:
   - Aktivieren Sie An Amazon CloudWatch Logs liefern. Wählen Sie die Zielprotokollgruppe aus.
   - Aktivieren Sie An Amazon S3 liefern. Geben Sie den Namen, den Besitzer und das Präfix des Ziel-Buckets ein.
   - Aktivieren Sie An Firehose liefern. Wählen Sie den Zielbereitstellungs-Stream aus.
7. Wählen Sie Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff ändern aus.

So aktivieren Sie Protokolle mit verifiziertem Zugriff mithilfe der AWS CLI

Verwenden Sie den Befehl modify-verified-access-instance-logging-configuration.

Deaktivieren der Zugriffsprotokolle

Sie können die Zugriffsprotokolle für Ihre Instance mit verifiziertem Zugriff jederzeit deaktivieren. Nachdem Sie die Zugriffsprotokolle deaktiviert haben, verbleiben Ihre Protokolldaten in Ihrem Protokollziel, bis Sie sie löschen.

So deaktivieren Sie Protokolle mit verifiziertem Zugriff

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Instances mit verifiziertem Zugriff aus.
3. Wählen Sie die Instance Verified Access aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff die Option Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff ändern aus.
5. Deaktivieren Sie die Protokollzustellung.
6. Wählen Sie Konfiguration der Instance-Protokollierung für verifizierten Zugriff ändern aus.

So deaktivieren Sie Protokolle mit verifiziertem Zugriff mithilfe der AWS CLI

Verwenden Sie den Befehl modify-verified-access-instance-logging-configuration.

Einschließen des Vertrauenskontexts

Der von Ihrem Vertrauensanbieter gesendete Vertrauenskontext kann optional in Ihre Protokolle für verifizierten Zugriff aufgenommen werden. Dies kann sehr nützlich sein, wenn Sie Richtlinien definieren, die den Zugriff auf Ihre Anwendungen zulassen oder verweigern. Nach der Aktivierung wird der Vertrauenskontext im Protokoll unter dem `data` Feld gefunden. Wenn diese Option deaktiviert ist, wird das `data` Feld auf gesetzt`null`. Gehen Sie wie folgt vor, um Verified Access so zu konfigurieren, dass Vertrauenskontext in die Protokolle aufgenommen wird.

> **Note**
>
> Das Einbinden von Vertrauenskontext in Ihre Protokolle für verifizierten Zugriff erfordert ein Upgrade auf die neueste Protokollierungsversion `ocsf-1.0.0-rc.2`. Im folgenden Verfahren wird davon ausgegangen, dass Sie die Protokollierung bereits aktiviert haben.

Wenn dies nicht zutrifft, finden Sie das vollständige Verfahren unter Aktivieren der Zugriffsprotokolle .

Inhalt

Aktivieren des Vertrauenskontexts

So schließen Sie den Vertrauenskontext mithilfe der Konsole in die Protokolle für verifizierten Zugriff ein

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Instances mit verifiziertem Zugriff aus.
3. Wählen Sie die entsprechende Instance mit verifiziertem Zugriff aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff die Option Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff ändern aus.
5. Wählen Sie ocsf-1.0.0-rc.2 aus der Dropdown-Liste Protokollversion aktualisieren aus.
6. Aktivieren Sie Vertrauenskontext einschließen.
7. Wählen Sie Konfiguration für die Instance-Protokollierung mit verifiziertem Zugriff ändern aus.

So schließen Sie den Vertrauenskontext mithilfe der in die Protokolle für verifizierten Zugriff ein AWS CLI

Verwenden Sie den Befehl modify-verified-access-instance-logging-configuration.

Deaktivieren des Vertrauenskontexts

Wenn Sie den Vertrauenskontext nicht mehr in die Protokolle aufnehmen möchten, können Sie ihn mit dem folgenden Verfahren entfernen.

So entfernen Sie den Vertrauenskontext aus den Protokollen für verifizierten Zugriff mithilfe der Konsole

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

2. Wählen Sie im Navigationsbereich Instances mit verifiziertem Zugriff aus.
3. Wählen Sie die entsprechende Instance mit verifiziertem Zugriff aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff die Option Konfiguration der Instance-Protokollierung mit verifiziertem Zugriff ändern aus.
5. Deaktivieren Sie Vertrauenskontext einschließen.
6. Wählen Sie Konfiguration für die Instance-Protokollierung mit verifiziertem Zugriff ändern aus.

So entfernen Sie den Vertrauenskontext aus den Protokollen für verifizierten Zugriff mithilfe der AWS CLI

Verwenden Sie den Befehl modify-verified-access-instance-logging-configuration.

Beispielprotokolleinträge für Protokolle mit verifiziertem Zugriff

Im Folgenden finden Sie Beispiele für Protokolleinträge.

Inhalt

Beispiele für O microSD Version 0.1

Im Folgenden finden Sie Beispielprotokolle, die die O microSD-Standardprotokollierung Version 0.1 verwenden.

Beispiele

Zugriff mit OIDC gewährt

In diesem Beispielprotokolleintrag ermöglicht Verified Access den Zugriff auf einen Endpunkt mit einem OIDC-Benutzervertrauensanbieter.


```
{
    "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
}
```

Zugriff mit OIDC und JAMF gewährt

In diesem Beispielprotokolleintrag ermöglicht Verified Access den Zugriff auf einen Endpunkt sowohl mit OIDC- als auch mit JAMF-Gerätevertrauensanbietern.


```
{
    "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
```

```
}
```

Zugriff gewährt mit OIDC und CrowdStrike

In diesem Beispielprotokolleintrag ermöglicht Verified Access den Zugriff auf einen Endpunkt sowohl mit OIDC CrowdStrike- als auch mit Gerätevertrauensanbietern.


```
{
    "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
}
```

Zugriff aufgrund eines fehlenden Cookies verweigert

In diesem Beispielprotokolleintrag verweigert Verified Access den Zugriff aufgrund eines fehlenden Authentifizierungscookie.

```
{
    "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
}
```

Zugriff durch Richtlinie verweigert

In diesem Beispielprotokolleintrag verweigert Verified Access eine authentifizierte Anforderung, da die Anforderung von den Zugriffsrichtlinien nicht zugelassen wird.

```
{
    "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
}
```


Unbekannter Protokolleintrag

In diesem Beispielprotokolleintrag kann Verified Access keinen vollständigen Protokolleintrag generieren, sodass es einen unbekannten Protokolleintrag ausgibt. Dadurch wird sichergestellt, dass jede Anforderung im Zugriffsprotokoll angezeigt wird.


```
{
    "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
}
```

Beispiele für O microSD Version 1.0.0-rc.2

Inhalt

Zugriff gewährt mit Vertrauenskontext enthalten

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

Zugriff ohne Vertrauenskontext gewährt


```
{
    "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
}
```

API-Aufrufe für AWS verifizierten Zugriff protokollieren mit AWS CloudTrail

AWSVerified Access ist mit integriertAWS CloudTrail, einem Service, der die Aktionen eines Benutzers, einer Rolle oder eines AWS-Service in Verified Access protokolliert. CloudTrailerfasst alle API-Aufrufe für Verified Network Manager als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Verified Access-Konsole und Code-Aufrufe der Verified Access-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon-S3-Bucket, einschließlich Ereignisse für Verified aktivieren. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole in Event history (Ereignisverlauf) anzeigen. Mit den von CloudTrail gesammelten Informationen können Sie die an Verified Access gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail-Benutzerhandbuch.

Verifizierte Zugangsinformationen in CloudTrail

CloudTrail wird beim Erstellen Ihres Kontos auf AWS-Konto aktiviert. Die in Verified Access auftretenden Aktivitäten werden als CloudTrail Ereignis zusammen mit anderen AWS-Service Ereignissen in Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr(em) AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail-Ereignisverlauf.

Erstellen Sie einen Trail zur laufenden Aufzeichnung der Ereignisse im -KontoAWS-Konto, einschließlich der Ereignisse für verifizierten Zugriff. Ein Trail ermöglicht esCloudTrail, Protokolldateien in einem Amazon-S3-Bucket bereitzustellen. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3 Bucket bereit. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Trails
- In CloudTrail unterstützte Services und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail

- Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail-Protokolldateien aus mehreren Konten

Alle Verified Accesseinträge werden von protokolliert CloudTrail und sind in der Amazon-EC2-API-Referenz dokumentiert. Zum Beispiel werden durch Aufrufe der Aktionen `CreateVerifiedAccessInstance`, `DeleteVerifiedAccessInstance` und `ModifyVerifiedAccessInstance` Einträge in den CloudTrail-Protokolldateien generiert.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM) -Anmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

Weitere Informationen finden Sie unter CloudTrail-Element userIdentity.

Grundlagen zu -Protokolldateieinträge

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar. Es enthält unter anderem Informationen über die angeforderte Aktion, etwaige Anforderungsparameter und das Datum und die Uhrzeit der Aktion. CloudTrail-Protokolleinträge sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt den CloudTrail-Protokolleintrag der Aktion `CreateVerifiedAccessInstance`.

```
{
    "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"
}
```

Kontingente für AWS verifizierten Zugriff

Ihr AWS-Konto verfügt über Standardkontingente, die früher als Limits bezeichnet wurden, für jedes KontingentAWS-Service. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region.

AWS-KontoKontingente auf zwei Ebenen

Ihr AWS-Konto hat die folgenden Kontingente in Bezug auf verifizierten Zugriff.

Name	Standard	Anpassbar	Beschreibung
Instanzen mit verifiziertem Zugriff	5	Ja	Die maximale Anzahl verifizierter Access-Instances, die Kunden in der aktuellen Region erstellen können.
Gruppen mit verifiziertem Zugriff	10	Ja	Die maximale Anzahl verifizierter Zugriffsgruppen, die Kunden in der aktuellen Region erstellen können.
Vertrauensanbieter mit verifiziertem Zugriff	15	Ja	Die maximale Anzahl verifizierter Access Trust Providers, die Kunden in der aktuellen Region einrichten können.
Verifizierte Zugriffse ndpunkte	50	Ja	Die maximale Anzahl verifizierter Zugriffsendpunkte, die Kunden in der aktuellen Region erstellen können.

HTTP-Header

Im Folgenden sind die Größenbeschränkungen für HTTP-Header aufgeführt.

Name	Standard	Anpassbar
Zeile anfordern	16 K	Nein
Einzelner Header	16 KM	Nein
Gesamter Antwort-Header	32 K	Nein
Gesamter Anforderungsheader	64 K	Nein

Größe des OIDC-Anspruchs

Im Folgenden ist die Obergrenze für die Größe eines OIDC-Anspruchs aufgeführt.

Name	Standard	Anpassbar
Größe des OIDC-Anspruchs	11 K	Nein

Dokumentenverlauf für das Verified Access-Benutzerhandbuch

In der folgenden Tabelle werden die Dokumentationsversionen für Verified Access beschrieben.

Änderung	Beschreibung	Datum
AWSDie verwaltete Richtlinie wurde aktualisiert	Die AWS verwaltete IAM-Richtlinie für verifizierten Zugriff wurde aktualisiert.	17. November 2023
Datenverschlüsselung im Ruhezustand	AWSVerified Access verschlüsselt Daten im Ruhezustand standardmäßig mithilfe AWS eigener KMS-Schlüssel.	28. September 2023
Unterstützung für FIPS-Compliance	Konfigurieren Sie Verified Access für FIPS-Konformität.	26. September 2023
Erweiterte Protokollierung	Hinzufügung einer Protokollierungsfunktion, die den Protokollen Vertrauenskontexte hinzufügt.	19. Juni 2023
AWSDie verwaltete Richtlinie wurde aktualisiert	Die AWS verwaltete IAM-Richtlinie für verifizierten Zugriff wurde aktualisiert.	31. Mai 2023
GA-Veröffentlichung	GA-Version des Verified Access-Benutzerhandbuchs. Beinhaltet AWS WAFIntegration.	27. April 2023
Vorschauversion	Vorschauversion des Verified Access-Benutzerhandbuchs	29. November 2022

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.