Benutzerhandbuch

AWS Verifizierter Zugriff

AWS Verifizierter Zugriff: Benutzerhandbuch

Table of Contents

Was ist AWS Verified Access?

Mit AWS Verified Access können Sie sicheren Zugriff auf Ihre Anwendungen bereitstellen, 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 bewertet jede Anwendungszugriffsanfrage in Echtzeit. Dies macht es für böswillige Akteure schwierig, 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.
- Verbessertes Benutzererlebnis — Verified Access macht es für Benutzer überflüssig, ein VPN für den Zugriff auf Ihre Anwendungen zu verwenden. Dies trägt dazu bei, die Anzahl der Supportfälle zu reduzieren, die sich aus VPN-Problemen ergeben.
- Vereinfachte Fehlerbehebung und Audits — Verified Access protokolliert alle Zugriffsversuche und bietet so einen zentralen Einblick in den Anwendungszugriff, sodass Sie schnell auf Sicherheitsvorfälle und Prüfanfragen reagieren können.

Zugriff auf Verified Access

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

- AWS Management Console— Stellt eine Weboberfläche bereit, mit der Sie verifizierte Access-Ressourcen erstellen und verwalten können. Melden Sie sich bei der an AWS Management Console 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 bereit AWS-Services, darunter AWS Verified Access. Das AWS CLI wird unter Windows, MacOS und

Linux unterstützt. Informationen zum Herunterladen AWS CLI finden Sie unter AWS Command Line Interface.

- AWS SDKs— Geben Sie sprachspezifisch APIs an. AWS SDKs Sie kümmern sich um viele Verbindungsdetails, z. B. um die Berechnung von Signaturen und die Bearbeitung von Wiederholungsversuchen und Fehlern bei Anfragen. 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 der direkteste Weg, um auf Verified Access zuzugreifen. Ihre Anwendung muss jedoch Details auf niedriger Ebene verarbeiten, z. B. die Generierung des Hashs zum Signieren der Anfrage und die Behandlung von Fehlern. Weitere Informationen finden Sie unter Aktionen mit verifiziertem Zugriff in der Amazon EC2 API-Referenz.

In diesem Handbuch wird beschrieben, wie Sie Ressourcen mit verifiziertem AWS Management Console Zugriff erstellen, darauf zugreifen und sie verwalten können.

Preisgestaltung

Für jede Anwendung auf Verified Access wird Ihnen stündlich eine Gebühr berechnet, und Ihnen wird die Menge der von Verified Access verarbeiteten Daten in Rechnung gestellt. Weitere Informationen finden Sie unter AWS Verified Access Preise.

So funktioniert Verified Access

AWS Verified Access bewertet jede Anwendungsanfrage Ihrer Benutzer und ermöglicht den Zugriff auf der Grundlage von:

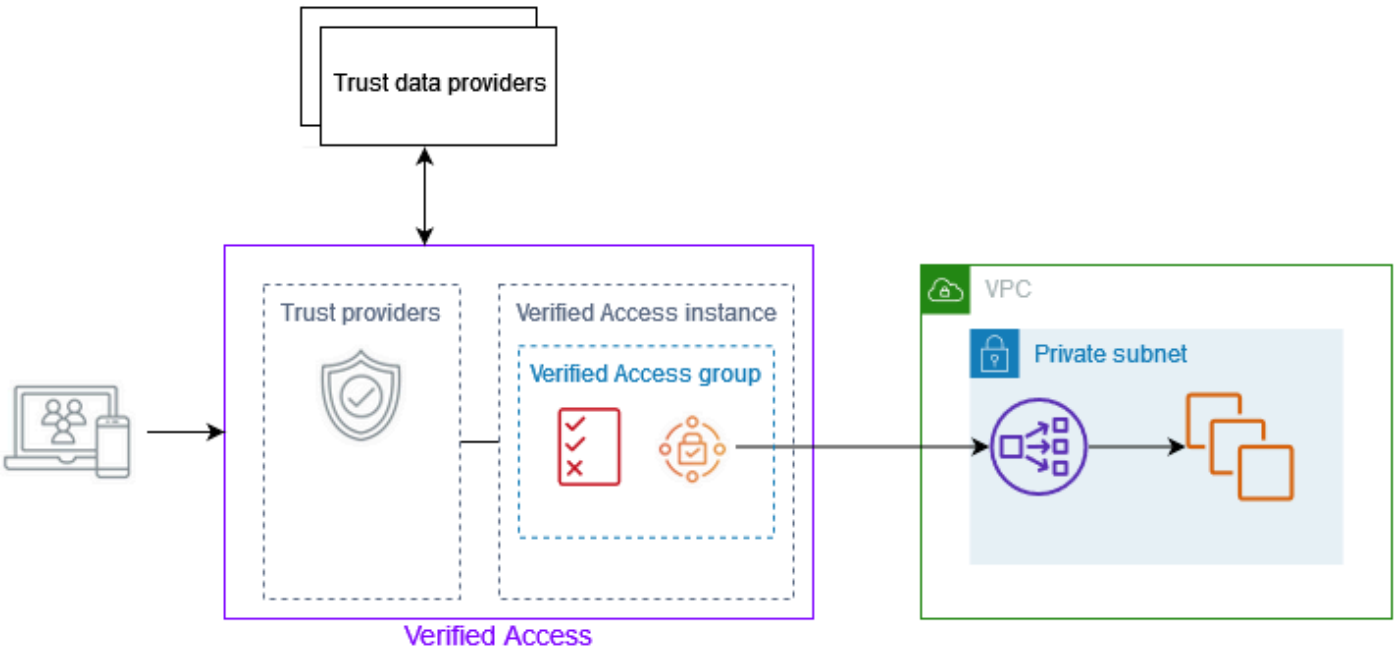
- Vertrauen Sie den Daten, die von Ihrem ausgewählten Vertrauensanbieter (von AWS oder einem Drittanbieter) gesendet wurden.
- Zugriffsrichtlinien, die Sie in Verified Access erstellen.

Wenn ein Benutzer versucht, auf eine Anwendung zuzugreifen, ruft Verified Access seine Daten vom Trust Provider ab und vergleicht sie mit den Richtlinien, 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 verweigert, bis eine Richtlinie definiert ist.

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

Die wichtigsten Komponenten von Verified Access

Das folgende Diagramm bietet einen allgemeinen Überblick über Verified Access. Benutzer senden Anfragen für den Zugriff auf eine Anwendung. 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.



- Verifizierte Zugriffsinstanzen — Eine Instanz bewertet Anwendungsanfragen und gewährt Zugriff nur, wenn Ihre Sicherheitsanforderungen erfüllt sind.
- Verifizierte Zugriffsendpunkte — Jeder Endpunkt steht für eine Anwendung. In der Abbildung oben wird die Anwendung auf EC2 Instanzen gehostet, die Ziele eines Load Balancers sind.
- Gruppe mit verifiziertem Zugriff — Eine Sammlung von Endpunkten mit verifiziertem Zugriff. 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 zu einer Gruppe zusammenfassen.
- Zugriffsrichtlinien — Eine Reihe von benutzerdefinierten Regeln, die festlegen, ob der Zugriff auf eine Anwendung zugelassen oder verweigert wird. Sie können eine Kombination von Faktoren angeben, darunter Benutzeridentität und Gerätesicherheitsstatus. 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.
- Vertrauensanbieter — Ein Dienst, der Benutzeridentitäten oder den Sicherheitsstatus von Geräten verwaltet. Verified Access funktioniert AWS sowohl mit vertrauenswürdigen Anbietern als auch mit Drittanbietern. 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 hinzufügen.

- 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 bewertet diese Daten anhand Ihrer Zugriffsrichtlinien, wenn es jede Anfrage zum Zugriff auf eine Anwendung erhält.

Tutorial: Erste Schritte mit Verified Access

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

Im Rahmen dieses Tutorials fügen Sie eine Anwendung zu Verified Access hinzu. Am Ende des Tutorials können bestimmte Benutzer über das Internet auf diese Anwendung zugreifen, ohne VPN zu verwenden. Stattdessen verwenden Sie es AWS IAM Identity Center als Identity Trust Provider. Beachten Sie, dass in diesem Tutorial nicht auch ein Device Trust Provider verwendet wird.

Aufgaben

Voraussetzungen für das Verified Access-Tutorial

Die folgenden Voraussetzungen müssen erfüllt sein, um dieses Tutorial abschließen zu können:

- AWS IAM Identity Center aktiviert in dem AWS-Region , in dem Sie gerade arbeiten. Anschließend können Sie IAM Identity Center als Vertrauensanbieter mit verifiziertem Zugriff verwenden. Weitere Informationen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
- Eine Sicherheitsgruppe zur Steuerung des Zugriffs auf die Anwendung. Lassen Sie den gesamten eingehenden Verkehr von der VPC CIDR und den gesamten ausgehenden Datenverkehr zu.
- Eine Anwendung, die hinter einem internen Load Balancer von Elastic Load Balancing ausgeführt wird. Ordnen Sie Ihre Sicherheitsgruppe dem Load Balancer zu.

- Ein selbstsigniertes oder öffentliches TLS-Zertifikat in. AWS Certificate Manager Verwenden Sie ein RSA-Zertifikat mit einer Schlüssellänge von 1.024 oder 2.048.
- Eine öffentlich gehostete Domain und die für die Aktualisierung der DNS-Einträge für die Domain erforderlichen Berechtigungen.
- Eine IAM-Richtlinie mit den zum Erstellen einer AWS Verified Access Instanz erforderlichen Berechtigungen. Weitere Informationen finden Sie unter Richtlinie für die Erstellung von Verified Access-Instanzen.

Schritt 1: Erstellen Sie einen vertrauenswürdigen Anbieter mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um ihn AWS IAM Identity Center als Ihren Vertrauensanbieter einzurichten.

So erstellen Sie einen IAM Identity Center Trust Provider

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Verified Access Trust Providers aus.
3. Wählen Sie Create Verified Access Trust Provider aus.
4. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Verified Access-Vertrauensanbieter ein.
5. Geben Sie einen benutzerdefinierten Bezeichner ein, der später bei der Arbeit mit Richtlinienregeln für den Referenznamen verwendet werden soll. Sie können beispielsweise eingeben**`idc`**.
6. Wählen Sie als Vertrauensanbietertyp die Option Benutzervertrauensdienstanbieter aus.
7. Wählen Sie als Typ des Vertrauensanbieters für Benutzer die Option IAM Identity Center aus.
8. Wählen Sie Create Verified Access Trust Provider aus.

Schritt 2: 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. Wählen Sie im Navigationsbereich Verified Access-Instanzen aus.

2. Wählen Sie Instanz mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie für Name und Beschreibung einen Namen und eine Beschreibung für die Verified Access-Instanz ein.
4. Wählen Sie für Verified Access Trust Provider Ihren Trust Provider aus.
5. Wählen Sie Create Verified Access-Instanz aus.

Schritt 3: Erstellen Sie eine Gruppe mit verifiziertem Zugriff

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

So erstellen Sie eine Gruppe mit verifiziertem Zugriff

1. Wählen Sie im Navigationsbereich Gruppen mit verifiziertem Zugriff aus.
2. Wählen Sie Gruppe mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für die Gruppe ein.
4. Wählen Sie für die Verified Access-Instanz Ihre Verified Access-Instanz aus.
5. Lassen Sie das Feld Richtliniendefinition leer. In einem späteren Schritt werden Sie eine Richtlinie auf Gruppenebene hinzufügen.
6. Wählen Sie Gruppe mit verifiziertem Zugriff erstellen aus.

Schritt 4: Erstellen Sie einen Endpunkt mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um einen Verified Access-Endpunkt zu erstellen. In diesem Schritt wird davon ausgegangen, dass Sie eine Anwendung hinter einem internen Load Balancer von Elastic Load Balancing und einem Public Domain-Zertifikat ausgeführt haben. AWS Certificate Manager

Um einen Verified Access-Endpunkt zu erstellen

1. Wählen Sie im Navigationsbereich die Option Verified Access-Endpoints aus.
2. Wählen Sie Endpunkt mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für den Endpunkt ein.
4. Wählen Sie für Gruppe mit verifiziertem Zugriff Ihre Gruppe mit verifiziertem Zugriff aus.

5. Gehen Sie wie folgt vor, um Endpunktdetails zu erhalten:

    a. Wählen Sie für Protokoll je nach Konfiguration Ihres Load Balancers HTTPS oder HTTP aus.

    b. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.

    c. Wählen Sie als Endpunkttyp die Option Load Balancer aus.

    d. Geben Sie für Port die Portnummer ein, die von Ihrem Load Balancer-Listener verwendet wird. Zum Beispiel 443 für HTTPS oder 80 für HTTP.

    e. Wählen Sie für Load Balancer ARN Ihren Load Balancer aus.

    f. Wählen Sie für Subnetze die Subnetze aus, die Ihrem Load Balancer zugeordnet sind.

    g. Wählen Sie für Sicherheitsgruppen Ihre Sicherheitsgruppe aus. Wenn Sie dieselbe Sicherheitsgruppe für Ihren Load Balancer und Ihren Endpunkt verwenden, ist Datenverkehr zwischen beiden möglich. Wenn Sie nicht dieselbe Sicherheitsgruppe verwenden möchten, stellen Sie sicher, dass Sie von Ihrem Load Balancer auf die Endpunkt-Sicherheitsgruppe verweisen, damit dieser den Datenverkehr vom Endpunkt akzeptiert.

    h. Geben Sie für Endpoint Domain Prefix eine benutzerdefinierte ID ein. Beispiel, **`my-ava-app`**. Dieses Präfix wird dem DNS-Namen vorangestellt, den Verified Access generiert.

6. Gehen Sie wie folgt vor, um Anwendungsdetails zu erhalten:

    a. Geben Sie unter Anwendungsdomäne den DNS-Namen für Ihre Anwendung ein. Diese Domain muss mit der Domain in Ihrem Domainzertifikat übereinstimmen.

    b. Wählen Sie unter Domainzertifikat ARN den Amazon-Ressourcennamen (ARN) Ihres Domainzertifikats aus AWS Certificate Manager.

7. Lassen Sie die Richtliniendetails leer. In einem späteren Schritt werden Sie eine Zugriffsrichtlinie auf Gruppenebene hinzufügen.

8. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Schritt 5: Konfigurieren Sie DNS für den Verified Access-Endpunkt

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

Um den Domainnamen Ihres Endpunkts abzurufen

1. Wählen Sie im Navigationsbereich Verified Access Endpoints aus.
2. Wählen Sie Ihren Endpunkt aus.
3. Wählen Sie die Registerkarte Details.
4. Kopieren Sie die Domäne aus der Endpunktdomäne. Im Folgenden finden Sie ein Beispiel für einen Endpunktdomänennamen:`my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`.

Folgen Sie den Anweisungen Ihres DNS-Anbieters, um einen CNAME-Eintrag zu erstellen. Verwenden Sie den Domainnamen Ihrer Anwendung als Datensatznamen und den Domainnamen des Verified Access-Endpunkts als Datensatzwert.

Schritt 6: Testen Sie die Konnektivität zur Anwendung

Sie können jetzt die Konnektivität zu Ihrer Anwendung testen. Geben Sie den Domainnamen Ihrer Anwendung in Ihren Webbrowser ein. Das Standardverhalten von Verified Access besteht darin, alle Anfragen abzulehnen. Da wir der Gruppe oder dem Endpunkt keine Richtlinie für verifizierten Zugriff hinzugefügt haben, werden alle Anfragen abgelehnt.

Schritt 7: Fügen Sie eine Zugriffsrichtlinie für verifizierten Zugriff auf Gruppenebene hinzu

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

Um eine Gruppe mit verifiziertem Zugriff zu ändern

1. Wählen Sie im Navigationsbereich Gruppen mit verifiziertem Zugriff aus.
2. Wählen Sie die -Gruppe aus.
3. Wählen Sie Aktionen, Gruppenrichtlinie für verifizierten Zugriff ändern aus.
4. Aktivieren Sie die Option „Richtlinie aktivieren“.

5. Geben Sie eine Richtlinie ein, die Benutzern von Ihrem IAM Identity Center den Zugriff auf Ihre Anwendung ermöglicht. Beispiele finden Sie unter the section called “Beispielrichtlinien”.
6. Wählen Sie Gruppenrichtlinie „Verifizierten Zugriff ändern“.
7. Nachdem Ihre Gruppenrichtlinie eingerichtet ist, wiederholen Sie den Test aus dem vorherigen Schritt, um sicherzustellen, dass die Anfrage zulässig ist. Wenn die Anfrage zulässig ist, werden Sie aufgefordert, sich über die IAM Identity Center-Anmeldeseite anzumelden. Nachdem Sie den Benutzernamen und das Passwort eingegeben haben, können Sie auf Ihre Anwendung zugreifen.

Bereinigen Sie Ihre Ressourcen für verifizierten Zugriff

Wenn Sie mit diesem Tutorial fertig sind, gehen Sie wie folgt vor, um Ihre Ressourcen mit verifiziertem Zugriff zu löschen.

So löschen Sie Ihre Ressourcen mit verifiziertem Zugriff

1. Wählen Sie im Navigationsbereich die Option Verified Access Endpoints aus. Wählen Sie den Endpunkt aus und klicken Sie auf Aktionen, Endpunkt mit verifiziertem Zugriff löschen.
2. Wählen Sie im Navigationsbereich Verifizierte Zugriffsgruppen aus. Wählen Sie die Gruppe aus und klicken Sie auf Aktionen, Gruppe mit verifiziertem Zugriff löschen. Möglicherweise müssen Sie warten, bis der Endpunkt-Löschvorgang abgeschlossen ist.
3. Wählen Sie im Navigationsbereich Verified Access-Instances aus. Wählen Sie Ihre Instance aus und klicken Sie dann auf Actions, Detach Verified Access Trust Provider. Wählen Sie den Vertrauensanbieter und anschließend den Vertrauensanbieter mit verifiziertem Zugriff trennen.
4. Wählen Sie im Navigationsbereich Verified Access Trust Providers aus. Wählen Sie Ihren Vertrauensanbieter aus und klicken Sie auf Aktionen, Vertrauensanbieter mit verifiziertem Zugriff löschen.
5. Wählen Sie im Navigationsbereich Verified Access-Instances aus. Wählen Sie Ihre Instanz aus und klicken Sie auf Aktionen, Instanz mit verifiziertem Zugriff löschen.

Verifizierte Access-Instanzen

Eine AWS Verified Access Instanz ist eine AWS Ressource, mit der Sie Ihre Vertrauensanbieter und Gruppen mit verifiziertem Zugriff organisieren können. Eine Instanz wertet Anwendungsanfragen aus und gewährt Zugriff nur, wenn Ihre Sicherheitsanforderungen erfüllt sind.

Aufgaben

- Erstellen und verwalten Sie eine Verified Access-Instanz
- Löschen Sie eine Instanz mit verifiziertem Zugriff
- Integrieren Sie Verified Access mit AWS WAF
- FIPS-Konformität für verifizierten Zugriff

Erstellen und verwalten Sie eine Verified Access-Instanz

Sie verwenden eine Verified Access-Instanz, um Ihre Vertrauensanbieter und Verified Access-Gruppen zu organisieren. Gehen Sie wie folgt vor, um eine Verified Access-Instanz zu erstellen und anschließend Verified Access einen Trust Provider zuzuordnen oder einen Trust Provider von Verified Access zu trennen.

Aufgaben

- Erstellen Sie eine Instanz mit verifiziertem Zugriff
- Ordnen Sie einer Verified Access-Instanz einen Vertrauensanbieter zu
- Trennen Sie einen Vertrauensanbieter von einer Verified Access-Instanz
- Fügen Sie eine benutzerdefinierte Subdomain hinzu

Erstellen Sie eine Instanz mit verifiziertem Zugriff

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

So erstellen Sie eine Verified Access-Instanz mithilfe der Konsole

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. (Netzwerk-CIDR-Endpunkte) Geben Sie für Benutzerdefinierte Subdomain für Netzwerk-CIDR-Endpoint eine benutzerdefinierte Subdomain ein.
5. (Optional) Wählen Sie Enable for Federal Information Process Standards (FIPS), wenn Verified Access FIPS-konform sein muss.
6. (Optional) Wählen Sie für Verified Access Trust Provider einen Trust Provider aus, der an die Verified Access-Instanz angehängt werden soll.
7. (Optional) Sie fügen ein Tag hinzu, indem Sie neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
8. Wählen Sie Instanz mit verifiziertem Zugriff erstellen aus.

Um eine Instanz mit verifiziertem Zugriff zu erstellen, verwenden Sie AWS CLI

Verwenden Sie den create-verified-access-instance-Befehl.

Ordnen Sie einer Verified Access-Instanz einen Vertrauensanbieter zu

Gehen Sie wie folgt vor, um einer Verified Access-Instanz einen Vertrauensanbieter zuzuordnen.

So fügen Sie mithilfe der Konsole einen Trust Provider an eine Verified Access-Instanz an

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.

Um einen Vertrauensanbieter an eine Verified Access-Instanz anzuhängen, verwenden Sie AWS CLI

Verwenden Sie den Befehl attach-verified-access-trust-provider.

Trennen Sie einen Vertrauensanbieter von einer Verified Access-Instanz

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

So trennen Sie mithilfe der Konsole einen Trust Provider 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.

Um einen Vertrauensanbieter von einer Verified Access-Instanz zu trennen, verwenden Sie AWS CLI

Verwenden Sie den Befehl detach-verified-access-trust-provider.

Fügen Sie eine benutzerdefinierte Subdomain hinzu

Gehen Sie wie folgt vor, um eine benutzerdefinierte Subdomain hinzuzufügen oder zu aktualisieren. Diese Subdomain wird nur verwendet, wenn Sie einen Netzwerk-CIDR-Endpunkt erstellen.

So fügen Sie mithilfe der Konsole eine benutzerdefinierte Subdomain 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, Instanz mit verifiziertem Zugriff ändern aus.
5. Geben Sie unter Benutzerdefinierte Subdomain für Netzwerk-CIDR-Endpunkt eine benutzerdefinierte Subdomain ein.
6. Wählen Sie Instanz mit verifiziertem Zugriff modifizieren aus.
7. Aktualisieren Sie die Nameserver für Ihre Subdomain und geben Sie die von Verified Access bereitgestellten Nameserver ein. Diese Liste ist unter Nameserver auf der Registerkarte Details für die Instanz verfügbar.

Um eine benutzerdefinierte Subdomain hinzuzufügen, verwenden Sie AWS CLI

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

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 mithilfe der Konsole zu löschen

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 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.

Um eine Instanz mit verifiziertem Zugriff zu löschen, verwenden Sie AWS CLI

Verwenden Sie den delete-verified-access-instance-Befehl.

Integrieren Sie Verified Access mit AWS WAF

Zusätzlich zu den Authentifizierungs- und Autorisierungsregeln, die von Verified Access durchgesetzt werden, möchten Sie möglicherweise auch den 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 WAF ist eine Firewall für Webanwendungen, mit der Sie die HTTP-Anfragen überwachen können, die an Ihre geschützten Webanwendungsressourcen weitergeleitet werden. Weitere Informationen finden Sie im AWS WAF - Entwicklerhandbuch.

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-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 beschreiben AWS CLI.

Der Vertrauensanbieter für Benutzeridentitäten bestimmt, wann der AWS WAF Datenverkehr überprüft wird. Wenn Sie IAM Identity Center verwenden, wird der Datenverkehr vor der AWS WAF Benutzerauthentifizierung überprüft. Wenn Sie OpenID Connect (OIDC) verwenden, AWS WAF überprüft es den Datenverkehr nach der Benutzerauthentifizierung.

Inhalt

Erforderliche IAM-Berechtigungen

Die Integration AWS WAF mit Verified Access umfasst Aktionen, die nur mit Zugriffsrechten ausgeführt werden 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 EC2 in der Service Authorization Reference unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon.

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 mithilfe der Verified Access-Konsole eine AWS WAF Web Access Control List (ACL) mit einer Verified Access-Instanz verknüpfen.

Voraussetzung

Bevor Sie beginnen, erstellen Sie eine AWS WAF Web-ACL. Weitere Informationen finden Sie unter Erstellen einer Web-ACL im AWS WAF Entwicklerhandbuch.

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.

Alternativ können Sie die AWS WAF Konsole verwenden. Wenn Sie die AWS WAF Konsole oder API verwenden, benötigen Sie den Amazon Resource Name (ARN) Ihrer Verified Access-Instance. Ein AVA-ARN hat das folgende Format:`arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}`. Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch unter Zuordnen einer Web-ACL zu einer AWS Ressource.

Überprüfen Sie den Status der Zuordnung

Mithilfe der Verified Access-Konsole können Sie überprüfen, ob eine AWS WAF Web Access Control List (ACL) mit einer Verified Access-Instanz verknüpft ist oder nicht.

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

In den folgenden Schritten wird veranschaulicht, wie Sie mithilfe der Verified Access-Konsole die Zuordnung einer AWS WAF Web Access Control List (ACL) zu einer Verified Access-Instanz aufheben.

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.

Alternativ können Sie die AWS WAF Konsole verwenden. Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch unter Trennen der Zuordnung einer Web-ACL zu einer AWS Ressource.

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 Access bietet 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)
- AWS GovCloud (US) Westen
- AWS GovCloud (US) Osten

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

Inhalt

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 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.

Vertraue Anbietern für verifizierten Zugriff

Ein Vertrauensanbieter ist ein Dienst, der Informationen über Benutzer und Geräte an sendet AWS Verified Access. 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

Vertrauenswürdige Anbieter von Benutzeridentitäten für verifizierten Zugriff

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.
- Verified Access kann Benutzern im IAM Identity Center, die bis zu 1.000 Gruppen zugewiesen sind, Zugriff gewähren.

Einzelheiten zu den verschiedenen Instanztypen finden Sie im AWS IAM Identity Center Benutzerhandbuch 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 neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
8. Wählen Sie Create Verified Access Trust Provider aus.

So erstellen Sie einen IAM Identity Center Trust Provider (AWS CLI)

- 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 (AWS Konsole) 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 das Löschen, indem Sie es `delete` in das Textfeld eingeben.
5. Wählen Sie Löschen aus.

So löschen Sie einen IAM Identity Center Trust Provider (AWS CLI)

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

Verwenden Sie einen OpenID Connect-Vertrauensanbieter

AWS Verified Access unterstützt Identitätsanbieter, die Standardmethoden OpenID Connect (OIDC) verwenden. Sie können OIDC-kompatible Anbieter als Vertrauensanbieter für Benutzeridentitäten mit verifiziertem Zugriff 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.

Die ID-Token-Ansprüche des OIDC-Vertrauensanbieters sind im `addition_user_context` Schlüssel für Vertrauensanbieter enthalten, die nach dem 24. Februar 2025 gegründet wurden.

Bei Vertrauensanbietern, die am oder vor dem 24. Februar 2025 gegründet wurden, verwendet Verified Access keine Vertrauensdaten aus den vom OIDC-Anbieter `ID token` gesendeten. Nur Vertrauensdaten von werden anhand `UserInfo Endpoint` der Richtlinie bewertet.

Die Sitzungsdauer für einen OIDC-Vertrauensanbieter beträgt 1 Tag oder die im Zugriffstoken angegebene Ablaufzeit, je nachdem, welcher Zeitraum kürzer ist. Bei Vertrauensanbietern, die vor dem 24. Februar 2025 eingerichtet wurden, beträgt die Sitzungsdauer 7 Tage.

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. Wählen Sie für OIDC (OpenID Connect) den Vertrauensanbieter aus.
8. Geben Sie für Issuer die ID des OIDC-Emittenten ein.
9. Geben Sie für Autorisierungsendpunkt die vollständige URL des Autorisierungsendpunkts ein.
10. Geben Sie für Token-Endpunkt die vollständige URL des Token-Endpunkts ein.
11. Geben Sie für Benutzerendpunkt die vollständige URL des Benutzerendpunkts ein.
12. (Native Application OIDC) Geben Sie für die URL des öffentlichen Signaturschlüssels die vollständige URL des Endpunkts mit dem öffentlichen Signaturschlüssel ein.
13. Geben Sie die OAuth 2.0-Client-ID als Client-ID ein.
14. Geben Sie für Client Secret den OAuth 2.0-Client-Schlüssel ein.
15. Geben Sie eine durch Leerzeichen getrennte Liste von Bereichen ein, die mit Ihrem Identitätsanbieter definiert wurden. Für openid Scope ist mindestens der Bereich erforderlich.
16. (Optional) Sie fügen ein Tag hinzu, indem Sie neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
17. Wählen Sie Create Verified Access Trust Provider aus.
18. Sie müssen der Zulassungsliste für Ihren OIDC-Anbieter eine Weiterleitungs-URI hinzufügen.
    - HTTP-Anwendungen — Verwenden Sie den folgenden URI:. **`https://application_domain/oauth2/idpresponse`** In der Konsole finden Sie die Anwendungsdomäne auf der Registerkarte Details für den Verified Access-Endpunkt. Wenn Sie das AWS CLI oder ein AWS SDK verwenden, ist die Anwendungsdomäne in der Ausgabe enthalten, wenn Sie den Verified Access-Endpunkt beschreiben.
    - TCP-Anwendungen — Verwenden Sie den folgenden URI:**`http://localhost:8000`**.

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 (AWS Konsole) 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 (AWS Konsole) 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 das Löschen, indem Sie es `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 für verifizierten Zugriff

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

Inhalt

Unterstützte Vertrauensanbieter für Geräte

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

- CrowdStrike — Sicherung privater Anwendungen mit CrowdStrike AWS 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 (AWS Konsole)

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 die Kennung 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 der AWS 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 (AWS CLI) 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 Vertrauensanbieter für Geräte mit verifiziertem Zugriff (AWS Konsole)

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 (AWS CLI) 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 (AWS Konsole)

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 (AWS CLI) mit verifiziertem Zugriff

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

Gruppen mit verifiziertem Zugriff

Eine Gruppe mit verifiziertem Zugriff besteht aus Endpunkten mit verifiziertem Zugriff und einer Richtlinie für verifizierten Zugriff, die für alle Endgeräte in der Gruppe gilt. Durch die Gruppierung von Endpunkten mit gemeinsamen Sicherheitsanforderungen können Sie eine einzelne Gruppenrichtlinie definieren, die die Mindestsicherheitsanforderungen mehrerer Endpunkte erfüllt. Daher müssen Sie nicht für jeden Endpunkt eine Richtlinie erstellen und verwalten.

Sie können beispielsweise alle Vertriebsanwendungen zusammenfassen 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 Richtlinienverwaltung bei.

Wenn Sie eine Gruppe erstellen, müssen Sie die Gruppe einer Verified Access-Instanz zuordnen. Während der Erstellung eines Endpoints ordnen Sie den Endpoint einer Gruppe zu.

Eine weitere Funktion von Gruppen mit verifiziertem Zugriff ist die Möglichkeit, sie mithilfe von anderen AWS Konten gemeinsam zu nutzen AWS RAM. Auf diese Weise können Sie Gruppen zentral in einem Konto erstellen und verwalten und sie dann mit mehreren Konten teilen.

Aufgaben

- Erstellen und verwalten Sie eine Gruppe mit verifiziertem Zugriff
- Ändern Sie eine Gruppenrichtlinie für verifizierten Zugriff
- Teilen Sie eine Gruppe mit verifiziertem Zugriff mit einer anderen AWS-Konto
- Löschen Sie eine Gruppe mit verifiziertem Zugriff

Erstellen und verwalten Sie eine Gruppe mit verifiziertem Zugriff

Sie verwenden Gruppen mit verifiziertem Zugriff, um Endgeräte nach ihren Sicherheitsanforderungen zu organisieren. Wenn Sie einen Verified Access-Endpunkt erstellen, ordnen Sie den Endpunkt einer Gruppe zu.

Aufgaben

- Erstellen Sie eine Gruppe mit verifiziertem Zugriff
- Ändern Sie eine Gruppe mit verifiziertem Zugriff

Erstellen Sie eine Gruppe mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um eine Gruppe mit verifiziertem Zugriff zu erstellen. Bevor Sie eine Verified Access-Gruppe erstellen, müssen Sie eine Verified Access-Instanz erstellen. Weitere Informationen finden Sie unter the section called “ Erstellen Sie eine Instanz mit verifiziertem Zugriff”.

Um eine Verified Access-Gruppe mithilfe der Konsole zu erstellen

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Gruppen mit verifiziertem Zugriff und anschließend Gruppe mit verifiziertem Zugriff erstellen aus.
3. (Optional) Geben Sie unter Namenstag und Beschreibung einen Namen und eine Beschreibung für die Gruppe ein.
4. Wählen Sie für die Instanz mit verifiziertem Zugriff eine Instanz mit verifiziertem Zugriff aus, die der Gruppe zugeordnet werden soll.
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 neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
7. Wählen Sie Gruppe mit verifiziertem Zugriff erstellen.

Um eine Gruppe mit verifiziertem Zugriff zu erstellen, verwenden Sie AWS CLI

Verwenden Sie den create-verified-access-group-Befehl.

Ändern Sie eine Gruppe mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um eine Gruppe mit verifiziertem Zugriff zu ändern.

So ändern Sie eine Gruppe mit verifiziertem Zugriff mithilfe der Konsole

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Gruppen mit verifiziertem Zugriff und anschließend Gruppe mit verifiziertem Zugriff erstellen aus.
3. Wählen Sie die Gruppe aus und klicken Sie dann auf Aktionen, Gruppe mit verifiziertem Zugriff ändern.

4. (Optional) Aktualisieren Sie die Beschreibung.
5. Wählen Sie Gruppe mit verifiziertem Zugriff erstellen.
6. Wählen Sie die Verified Access-Instanz aus, die Sie der Gruppe zuordnen möchten.

Um eine Verified Access-Gruppe mit dem AWS CLI

Verwenden Sie den modify-verified-access-group-Befehl.

Ändern Sie eine Gruppenrichtlinie für verifizierten Zugriff

AWS Verified Access ermöglicht den Zugriff auf Ihre Anwendungen auf der Grundlage der von Ihnen erstellten Zugriffsrichtlinien. Die Verified Access-Richtlinie, die Sie einer Gruppe zuordnen, wird von allen Endpunkten in der Gruppe übernommen. Sie können optional anwendungsspezifische Richtlinien an bestimmte Endpunkte anhängen.

Gehen Sie wie folgt vor, um die Richtlinie für eine Gruppe mit verifiziertem Zugriff zu ändern. Nachdem Sie die Änderungen vorgenommen haben, dauert es einige Minuten, bis sie wirksam werden.

So ändern Sie eine Gruppenrichtlinie 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 Gruppen mit verifiziertem Zugriff aus.
3. Wählen Sie die -Gruppe aus.
4. Wählen Sie Aktionen, Gruppenrichtlinie für verifizierten Zugriff ändern aus.
5. (Optional) Aktivieren oder deaktivieren Sie die Option „Richtlinie aktivieren“ nach Bedarf.
6. (Optional) Geben Sie unter Richtlinie die Richtlinie für verifizierten Zugriff ein, die auf die Gruppe angewendet werden soll.
7. Wählen Sie Gruppenrichtlinie „Verifizierten Zugriff ändern“ aus.

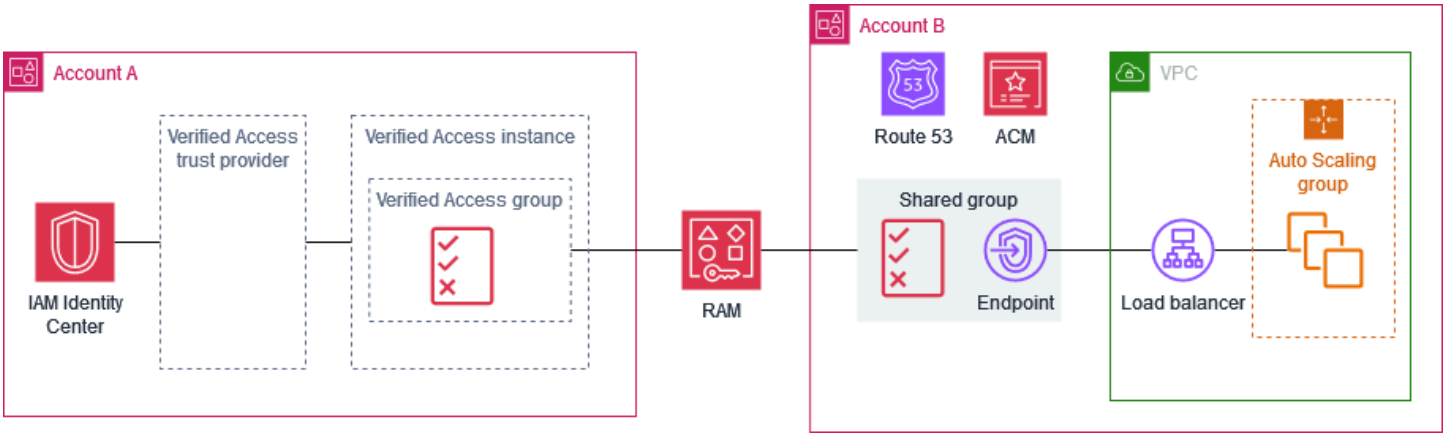
Um eine Gruppenrichtlinie mit verifiziertem Zugriff zu ändern, verwenden Sie AWS CLI

Verwenden Sie den Befehl modify-verified-access-group-policy.

Teilen Sie eine Gruppe mit verifiziertem Zugriff mit einer anderen AWS-Konto

Wenn Sie eine Gruppe mit verifiziertem Zugriff, deren Inhaber Sie sind, mit anderen AWS Konten teilen, ermöglichen Sie diesen Konten, Endpunkte mit verifiziertem Zugriff in Ihrer Gruppe zu erstellen. Das Konto, in dem die Gruppe mit verifiziertem Zugriff erstellt wurde, wird als Besitzerkonto bezeichnet. Das Konto, das eine gemeinsam genutzte Gruppe verwendet, wird als Verbraucherkonto bezeichnet.

Das folgende Diagramm veranschaulicht die Vorteile der gemeinsamen Nutzung einer Gruppe mit verifiziertem Zugriff. Das zentrale Sicherheitsteam ist für Konto A verantwortlich. Es verwaltet Benutzer und Gruppen sowie die Ressourcen für verifizierten Zugriff AWS IAM Identity Center, die für den Zugriff auf interne Anwendungen erforderlich sind, wie z. B. Vertrauensanbieter mit verifiziertem Zugriff, Instanzen mit verifiziertem Zugriff, Gruppen mit verifiziertem Zugriff und Richtlinien für verifizierten Zugriff. Das Anwendungsteam besitzt Account B. Es verwaltet die Ressourcen, die für die Ausführung der internen Anwendung erforderlich sind, wie z. B. den Load Balancer, die Auto Scaling Scaling-Gruppe, die DNS-Konfiguration in Amazon Route 53 und die TLS-Zertifikate von AWS Certificate Manager (ACM). Nachdem das zentrale Sicherheitsteam eine Verified Access-Gruppe mit Konto B gemeinsam genutzt hat, kann das Anwendungsteam mithilfe der gemeinsamen Gruppe Verified Access-Endpunkte erstellen. Der Zugriff auf die Anwendung wird auf der Grundlage der Richtlinien, die das zentrale Sicherheitsteam für die Verified Access-Gruppe erstellt hat, erlaubt oder verweigert.



Überlegungen

Die folgenden Überlegungen gelten für gemeinsam genutzte Gruppen mit verifiziertem Zugriff.

Eigentümer

- Um eine Gruppe mit verifiziertem Zugriff gemeinsam nutzen zu können, müssen Benutzer über die folgenden Berechtigungen verfügen: `ec2:PutResourcePolicy` und`ec2:DeleteResourcePolicy`.
- Um eine Gruppe mit verifiziertem Zugriff teilen zu können, müssen Sie Eigentümer dieser Gruppe sein. Sie können eine Gruppe mit verifiziertem Zugriff, die mit Ihnen geteilt wurde, nicht teilen.
- Wenn Sie das Teilen mit den Konten in Ihrer Organisation aktivieren, können Sie Ressourcen wie Gruppen mit verifiziertem Zugriff gemeinsam nutzen, ohne Einladungen zu verwenden. Andernfalls erhält der Verbraucher eine Einladung und muss sie annehmen, um auf die gemeinsam genutzte Gruppe zugreifen zu können. Um das Teilen zu aktivieren, öffnen Sie vom Verwaltungskonto Ihrer Organisation aus die Seite Einstellungen in der AWS RAM Konsole und wählen Sie Teilen aktivieren mit aus AWS Organizations.
- Sie können eine Gruppe nicht löschen, wenn ihnen Verified Access-Endpunkte zugeordnet sind. Sie können die von Kundenkonten erstellten Endpunkte auf der Seite mit verifiziertem Zugriff in Ihrem Konto einsehen. Die Konto-ID des Besitzers eines Endpunkts spiegelt sich im Amazon-Ressourcennamen (ARN) des Zertifikats für den Endpunkt wider.

Konsumenten

- Um sich die Gruppen mit verifiziertem Zugriff anzusehen, die mit Ihnen geteilt wurden, öffnen Sie die Seite mit verifiziertem Zugriff in der Konsole oder rufen Sie uns an describe-verified-access-groups. Die Konto-ID des Besitzers wird im Feld Besitzer und im Amazon-Ressourcennamen (ARN) der Gruppe wiedergegeben.
- Wenn Sie einen Endpunkt mit verifiziertem Zugriff erstellen, können Sie alle Gruppen mit verifiziertem Zugriff angeben, die für Sie freigegeben wurden.
- Sie können keine Endpoints anzeigen, die der gemeinsam genutzten Gruppe zugeordnet sind, aber nicht Ihnen gehören.
- Wenn der Besitzer der Gruppe mit verifiziertem Zugriff die Ressourcenfreigabe löscht, können Sie in der Gruppe keinen neuen Endpunkt mit verifiziertem Zugriff erstellen. Alle Endpunkte mit verifiziertem Zugriff, die Sie vor dem Löschen der Ressourcenfreigabe erstellt haben, sind von der Löschung der Ressourcenfreigabe nicht betroffen. Der Besitzer der gemeinsam genutzten Gruppe kann Ihre Endpoints jedoch löschen.

Gemeinsam genutzte Ressourcen

Um eine Gruppe mit verifiziertem Zugriff gemeinsam zu nutzen, müssen Sie sie zu einer Ressourcenfreigabe hinzufügen. Eine Ressourcenfreigabe gibt die Ressourcen an, die gemeinsam genutzt werden sollen, und die Benutzer, die die gemeinsam genutzten Ressourcen nutzen können.

Um eine Gruppe mit verifiziertem Zugriff über die Konsole gemeinsam zu nutzen

1. Öffnen Sie die AWS RAM Konsole zu https://console.aws.amazon.com/ram/Hause.
2. Wenn Sie noch keine Ressourcenfreigabe für Ihre Organisation haben, erstellen Sie eine. Für den Prinzipal können Sie Ihre gesamte Organisation, eine Organisationseinheit oder bestimmte AWS Konten auswählen.
3. Wählen Sie Ihren Ressourcenanteil aus und klicken Sie auf Ändern.
4. Wählen Sie für `Resources` Verifizierte Zugriffsgruppen als Ressourcentyp und wählen Sie dann die Ressourcengruppe aus, die Sie teilen möchten.
5. Wählen Sie „Direkt zu: Überprüfen und aktualisieren“.
6. Wählen Sie „Ressourcenfreigabe aktualisieren“.

Weitere Informationen finden Sie unter Erstellen einer Ressourcenfreigabe im AWS RAM - Benutzerhandbuch.

Löschen Sie eine Gruppe mit verifiziertem Zugriff

Wenn Sie mit einer Gruppe mit verifiziertem Zugriff fertig sind, können Sie sie löschen. Sie können eine Gruppe nicht löschen, wenn ihr Endpunkte mit verifiziertem Zugriff zugeordnet sind.

Um eine Gruppe mit verifiziertem Zugriff mithilfe der Konsole zu löschen

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

Um eine Gruppe mit verifiziertem Zugriff zu löschen, verwenden Sie AWS CLI

Verwenden Sie den delete-verified-access-group-Befehl.

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 hinzufügen.

Inhalt

Verifizierte Access-Endpunkttypen

Im Folgenden sind die möglichen Endpunkttypen für verifizierten Zugriff aufgeführt:

- Load Balancer — Anwendungsanfragen werden an einen Load Balancer gesendet, um sie an Ihre Anwendung zu verteilen. Weitere Informationen finden Sie unter Erstellen Sie einen Load Balancer-Endpunkt.
- Netzwerkschnittstelle — Anwendungsanfragen werden unter Verwendung des angegebenen Protokolls und Ports an eine Netzwerkschnittstelle gesendet. Weitere Informationen finden Sie unter Erstellen Sie einen Netzwerkschnittstellen-Endpunkt.
- Netzwerk-CIDR — Anwendungsanfragen werden an den angegebenen CIDR-Block gesendet. Weitere Informationen finden Sie unter Erstellen Sie einen Netzwerk-CIDR-Endpunkt.
- Amazon Relational Database Service (RDS) — Anwendungsanfragen werden an eine RDS-Instance, einen RDS-Cluster oder einen RDS-DB-Proxy gesendet. Weitere Informationen finden Sie unter Einen Amazon Relational Database Service Service-Endpunkt erstellen.

So funktioniert Verified Access mit geteilten Netzen VPCs und Subnetzen

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 für Verified Access zu erstellen. Weitere Informationen zu Load Balancern finden Sie im Elastic Load Balancing User Guide.

Voraussetzungen

- Es wird nur IPv4 Traffic unterstützt.
- Langlebige HTTPS-Verbindungen, wie z. B. WebSocket Verbindungen, werden nur über TCP 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. Verwenden Sie ein RSA-Zertifikat mit einer Schlüssellänge von 1.024 oder 2.048.
- Bevor Sie einen Verified Access-Endpunkt erstellen, müssen Sie eine Verified Access-Gruppe erstellen. Weitere Informationen finden Sie unter the section called “Erstellen Sie eine Gruppe mit verifiziertem Zugriff”.

- Sie müssen einen Domänennamen 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 importieren AWS Certificate Manager.

Um einen Load Balancer-Endpunkt mit der Konsole 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 aus.
6. Gehen Sie wie folgt vor, um Endpunktdetails zu erhalten:
   a. Wählen Sie unter Protokoll ein Protokoll aus.
   b. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.
   c. Wählen Sie als Endpunkttyp die Option Load Balancer aus.
   d. (HTTP/HTTPS) Geben Sie für Port die Portnummer ein. (TCP) Geben Sie für Portbereiche einen Portbereich ein und wählen Sie Port hinzufügen aus.
   e. Wählen Sie für Load Balancer ARN einen Load Balancer aus.
   f. Wählen Sie für Subnetz die Subnetze aus. Sie können nur ein Subnetz pro Availability Zone angeben.
   g. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppen für den Endpunkt aus. Diese Sicherheitsgruppen kontrollieren den eingehenden und ausgehenden Datenverkehr für den Verified Access-Endpunkt.
   h. Geben Sie für das Endpunktdomänenpräfix einen benutzerdefinierten Bezeichner ein, der dem DNS-Namen vorangestellt wird, den Verified Access für den Endpunkt generiert.
7. (HTTP/HTTPS) 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 ein öffentliches TLS-Zertifikat aus.
8. (Optional) Geben Sie für die Richtliniendefinition eine verifizierte Zugriffsrichtlinie für den Endpunkt ein.

9. (Optional) Sie fügen ein Tag hinzu, indem Sie neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
10. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Um einen Verified Access-Endpunkt mit dem zu erstellen AWS CLI

Verwenden Sie den create-verified-access-endpoint-Befehl.

Erstellen Sie einen Netzwerkschnittstellen-Endpunkt für Verified Access

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

Voraussetzungen

- Es wird nur IPv4 Verkehr 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.
- Bevor Sie einen Verified Access-Endpunkt erstellen, müssen Sie eine Verified Access-Gruppe erstellen. Weitere Informationen finden Sie unter the section called “Erstellen Sie eine Gruppe mit verifiziertem Zugriff”.
- Sie müssen einen Domänennamen 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 importieren AWS Certificate Manager.

Um einen Netzwerkschnittstellen-Endpunkt mit der Konsole 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 aus.

6. Gehen Sie wie folgt vor, um Endpunktdetails zu erhalten:

    a. Wählen Sie unter Protokoll ein Protokoll aus.

    b. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.

    c. Wählen Sie als Endpunkttyp die Option Netzwerkschnittstelle aus.

    d. (HTTP/HTTPS) Geben Sie für Port die Portnummer ein. (TCP) Geben Sie für Portbereiche einen Portbereich ein und wählen Sie Port hinzufügen aus.

    e. Wählen Sie für Netzwerkschnittstelle eine Netzwerkschnittstelle aus.

    f. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppen für den Endpunkt aus. Diese Sicherheitsgruppen kontrollieren den eingehenden und ausgehenden Datenverkehr für den Verified Access-Endpunkt.

    g. Geben Sie für das Endpunktdomänenpräfix einen benutzerdefinierten Bezeichner ein, der dem DNS-Namen vorangestellt wird, den Verified Access für den Endpunkt generiert.

7. (HTTP/HTTPS) 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 ein öffentliches TLS-Zertifikat aus.

8. (Optional) Geben Sie für die Richtliniendefinition eine verifizierte Zugriffsrichtlinie für den Endpunkt ein.

9. (Optional) Sie fügen ein Tag hinzu, indem Sie neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.

10. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Um einen Verified Access-Endpunkt mit dem zu erstellen AWS CLI

Verwenden Sie den create-verified-access-endpoint-Befehl.

Erstellen Sie einen Netzwerk-CIDR-Endpunkt für Verified Access

Gehen Sie wie folgt vor, um einen Netzwerk-CIDR-Endpunkt zu erstellen. Sie können beispielsweise einen Netzwerk-CIDR-Endpunkt verwenden, um den Zugriff auf EC2 Instanzen in einem bestimmten Subnetz über Port 22 (SSH) zu ermöglichen.

Voraussetzungen

- Nur das TCP-Protokoll wird unterstützt.

- Verified Access stellt einen DNS-Eintrag für jede IP-Adresse im CIDR-Bereich bereit, die von einer Ressource verwendet wird. Wenn Sie eine Ressource löschen, wird ihre IP-Adresse nicht mehr verwendet und Verified Access löscht den entsprechenden DNS-Eintrag.
- Wenn Sie eine benutzerdefinierte Subdomain angeben, stellt Verified Access DNS-Einträge für jede IP-Adresse bereit, die in der Subdomain verwendet wird, und stellt Ihnen die IP-Adressen der zugehörigen DNS-Server zur Verfügung. Sie können eine Weiterleitungsregel für Ihre Subdomain so konfigurieren, dass sie auf die DNS-Server mit verifiziertem Zugriff verweist. Jede Anfrage an einen Datensatz in der Domain wird von den Verified Access-DNS-Servern an die IP-Adresse der angeforderten Ressource aufgelöst.
- Bevor Sie einen Verified Access-Endpunkt erstellen, müssen Sie eine Verified Access-Gruppe erstellen. Weitere Informationen finden Sie unter the section called “Erstellen Sie eine Gruppe mit verifiziertem Zugriff”.
- Erstellen Sie den Endpunkt und stellen Sie dann über den eine Verbindung mit der Anwendung herKonnektivitätsclient.

Um einen Netzwerk-CIDR-Endpunkt mit der Konsole 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 Endpunktdetails zu erhalten:

    a. Wählen Sie für Protocol TCP aus.

    b. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.

    c. Wählen Sie als Endpunkttyp Network CIDR aus.

    d. Geben Sie für Portbereiche einen Portbereich ein und wählen Sie Port hinzufügen aus.

    e. Wählen Sie für Subnetz die Subnetze aus.

    f. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppen für den Endpunkt aus. Diese Sicherheitsgruppen kontrollieren den eingehenden und ausgehenden Datenverkehr für den Verified Access-Endpunkt.

g. (Optional) Geben Sie für das Endpoint-Domänenpräfix einen benutzerdefinierten Bezeichner ein, der dem DNS-Namen vorangestellt wird, den Verified Access für den Endpunkt generiert.

7. (Optional) Geben Sie für die Richtliniendefinition eine verifizierte Zugriffsrichtlinie für den Endpunkt ein.
8. (Optional) Sie fügen ein Tag hinzu, indem Sie neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
9. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Um einen Verified Access-Endpunkt mit dem zu erstellen AWS CLI

Verwenden Sie den create-verified-access-endpoint-Befehl.

Erstellen Sie einen Amazon Relational Database Service Service-Endpunkt für verifizierten Zugriff

Gehen Sie wie folgt vor, um einen Amazon Relational Database Service (RDS) -Endpunkt zu erstellen.

Voraussetzungen

- Nur das TCP-Protokoll wird unterstützt.
- Erstellen Sie eine RDS-Instanz, einen RDS-Cluster oder einen RDS-DB-Proxy.
- Bevor Sie einen Verified Access-Endpunkt erstellen, müssen Sie eine Verified Access-Gruppe erstellen. Weitere Informationen finden Sie unter the section called “Erstellen Sie eine Gruppe mit verifiziertem Zugriff”.
- Erstellen Sie den Endpunkt und stellen Sie dann über den eine Verbindung mit der Anwendung herKonnektivitätsclient.

So erstellen Sie mit der Konsole einen Amazon Relational Database Service Service-Endpunkt

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 Endpunktdetails zu erhalten:
   a. Wählen Sie für Protocol TCP aus.
   b. Wählen Sie bei Attachment type (Anhangstyp) die Option VPC aus.
   c. Wählen Sie als Endpunkttyp Amazon Relational Database Service (RDS) aus.
   d. Führen Sie für den RDS-Zieltyp einen der folgenden Schritte aus:
      - Wählen Sie RDS-Instance und dann eine RDS-Instance aus der RDS-Instance aus.
      - Wählen Sie RDS-Cluster und dann einen RDS-Cluster aus dem RDS-Cluster aus.
      - Wählen Sie RDS-DB-Proxy und dann einen RDS-DB-Proxy aus dem RDS-DB-Proxy.
   e. Wählen Sie für den RDS-Endpunkt einen RDS-Endpunkt aus, der sich auf die RDS-Ressource bezieht, die Sie im vorherigen Schritt ausgewählt haben.
   f. Geben Sie unter Port die Portnummer ein.
   g. Wählen Sie für Subnetz die Subnetze aus. Sie können nur ein Subnetz pro Availability Zone angeben.
   h. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppen für den Endpunkt aus. Diese Sicherheitsgruppen kontrollieren den eingehenden und ausgehenden Datenverkehr für den Verified Access-Endpunkt.
   i. (Optional) Geben Sie für das Endpoint-Domänenpräfix einen benutzerdefinierten Bezeichner ein, der dem DNS-Namen vorangestellt wird, den Verified Access für den Endpunkt generiert.
7. (Optional) Geben Sie für die Richtliniendefinition eine verifizierte Zugriffsrichtlinie für den Endpunkt ein.
8. (Optional) Sie fügen ein Tag hinzu, indem Sie neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
9. Wählen Sie „Verifizierten Zugriffsendpunkt erstellen“.

Um einen Verified Access-Endpunkt mit dem zu erstellen AWS CLI

Verwenden Sie den create-verified-access-endpoint-Befehl.

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 mithilfe der Konsole 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 IDs auf der Registerkarte Details nach Sicherheitsgruppe 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 Regel hinzufügen aus.
   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.

Um die Sicherheitsgruppenregeln für Ihre Anwendung mit dem zu aktualisieren AWS CLI

Verwenden Sie den describe-verified-access-endpointsBefehl, um die ID der Sicherheitsgruppe abzurufen, und fügen Sie dann mit dem authorize-security-group-ingressBefehl eine Regel für eingehenden Datenverkehr hinzu.

Ändern Sie einen Endpunkt mit verifiziertem Zugriff

Gehen Sie wie folgt vor, um einen Verified Access-Endpunkt zu ändern.

So ändern Sie einen Verified Access-Endpunkt mithilfe der Konsole

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.

Um einen Endpunkt mit verifiziertem Zugriff zu ändern, verwenden Sie den AWS CLI

Verwenden Sie den modify-verified-access-endpoint-Befehl.

Ändern Sie eine Endpunktrichtlinie für verifizierten Zugriff

Gehen Sie wie folgt vor, um die Richtlinie für einen Endpunkt mit verifiziertem Zugriff zu ändern. Nachdem Sie die Änderungen vorgenommen haben, dauert es einige Minuten, bis sie wirksam werden.

So ändern Sie eine Endpunktrichtlinie 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 die Option Verified Access Endpoints aus.
3. Wählen Sie den Endpunkt.
4. Wählen Sie Aktionen, Endpunktrichtlinie für verifizierten Zugriff ändern aus.
5. (Optional) Aktivieren oder deaktivieren Sie die Option Richtlinie aktivieren nach Bedarf.
6. (Optional) Geben Sie unter Policy die Verified Access-Richtlinie ein, die auf den Endpunkt angewendet werden soll.

7. Wählen Sie Endpunktrichtlinie für verifizierten Zugriff ändern aus.

Um eine Endpunktrichtlinie für verifizierten Zugriff zu ändern, verwenden Sie AWS CLI

Verwenden Sie den Befehl modify-verified-access-endpoint-policy.

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 mit der Konsole 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.

Um einen Endpunkt mit verifiziertem Zugriff zu löschen, verwenden Sie AWS CLI

Verwenden Sie den delete-verified-access-endpoint-Befehl.

Vertrauensdaten, die von Vertrauensanbietern an Verified Access gesendet wurden

Vertrauensdaten sind Daten, AWS Verified Access an die ein Vertrauensdienstanbieter gesendet wird. Vertrauensdaten werden auch als „Benutzeransprüche“ oder „Vertrauenskontext“ bezeichnet. Die Daten umfassen 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 usw. 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 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 unterAktivieren oder deaktivieren Sie den Vertrauenskontext Verified Access.

Dieser Abschnitt enthält Beispiele für Vertrauensdaten und Beispiele, die Ihnen den Einstieg in die Erstellung von Richtlinien erleichtern sollen. Die hier bereitgestellten Informationen dienen nur zur Veranschaulichung und nicht als offizielle Referenz.

Inhalt

Standardkontext für Vertrauensdaten mit verifiziertem Zugriff

AWS Verified Access enthält standardmäßig einige Elemente zur aktuellen Anfrage in allen Cedar-Evaluierungen, unabhängig von Ihren konfigurierten Vertrauensanbietern. Sie können eine Richtlinie schreiben, die anhand der Daten bewertet wird, wenn Sie möchten.

Im Folgenden finden Sie Beispiele für die Daten, die in der Auswertung enthalten sind.

Beispiele

- HTTP-Anfrage
- TCP-Fluss

HTTP-Anfrage

Wenn eine Richtlinie ausgewertet wird, enthält Verified Access unter dem `context.http_request` Schlüssel Daten über die aktuelle HTTP-Anfrage im Cedar-Kontext.

```
{
    "title": "HTTP Request data included by Verified Access",
    "type": "object",
    "properties": {
        "http_method": {
            "type": "string",
            "description": "The HTTP method",
            "example": "GET"
        },
        "hostname": {
            "type": "string",
            "description": "The host subcomponent of the authority component of the
 URI",
            "example": "example.com"
        },
        "path": {
            "type": "string",
            "description": "The path component of the URI",
            "example": "app/images"
        },
        "query": {
            "type": "string",
            "description": "The query component of the URI",
            "example": "value1=1&value2=2"
        },
        "x_forwarded_for": {
            "type": "string",
            "description": "The value of the X-Forwarded-For request header",
            "example": "17.7.7.1"
        },
        "port": {
           "type": "integer",
           "description": "The endpoint port",
           "example": 443
```

```
        },
        "user_agent": {
            "type": "string",
            "description": "The value of the User-Agent request header",
            "example": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0)
 Gecko/20100101 Firefox/47.0"
        },
        "client_ip": {
            "type": "string",
            "description": "The IP address connecting to the endpoint",
            "example": "15.248.6.6"
        }
    }
}
```

Beispiele für Richtlinien

Im Folgenden finden Sie ein Beispiel für eine Cedar-Richtlinie, die die HTTP-Anforderungsdaten verwendet.

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

TCP-Fluss

Wenn eine Richtlinie ausgewertet wird, enthält Verified Access unter dem `context.tcp_flow` Schlüssel Daten über den aktuellen TCP-Fluss im Cedar-Kontext.

```
{
    "title": "TCP flow data included by Verified Access",
    "type": "object",
    "properties": {
        "destination_ip": {
            "type": "string",
            "description": "The IP address of the target",
            "example": "192.100.1.3"
        },
        "destination_port": {
            "type": "string",
            "description": "The target port",
```

```
            "example": 22
        },
        "client_ip": {
            "type": "string",
            "description": "The IP address connecting to the endpoint",
            "example": "172.154.16.9"
        }
    }
}
```

AWS IAM Identity Center Kontext für Vertrauensdaten von Verified Access

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

> **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.

Kontext eines Drittanbieter-Vertrauensanbieters für Vertrauensdaten mit verifiziertem Zugriff

In diesem Abschnitt werden die Vertrauensdaten beschrieben, die AWS Verified Access von externen Vertrauensanbietern 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 Nutzer die AWS Verified Access Browsererweiterung vom Chrome Web Store oder von der Firefox Add-On-Website auf ihren Geräten herunterladen und installieren.
- Wenn Sie CrowdStrikeVertrauensdaten in Ihren Richtlinien verwenden, müssen Ihre Benutzer zunächst den AWS Verified Access Native Messaging Host installieren (direkter Download-Link). 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 AWS Verified Access Browsererweiterung aus dem Chrome-Webshop oder der Firefox-Add-On-Website auf ihren Geräten installieren.
- Wenn 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 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 Verified Access finden Sie unter Schützen privater Anwendungen mit CrowdStrike und AWS Verified Access auf der GitHub Website.

```
{
  "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 environment"
    },
    "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_organization_identifier'
};
```

Der Benutzer gibt in Verified Access die Weitergabe und Überprüfung der Signatur an

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 die Signaturen überprüfen und 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"
   ],
   "additional_user_context": {
      "aud": "xxx",
      "exp": 1000000000,
      "groups": [
         "group-id-1",
         "group-id-2"
      ],
      "iat": 1000000000,
      "iss": "https://oidc-tp.com/",
      "sub": "xyzsubject",
      "ver": "1.0"
   }
}
```

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.

Um die Sicherheit zu gewährleisten, müssen Sie die Signatur überprüfen, bevor Sie eine Autorisierung auf der Grundlage der Ansprüche durchführen, und überprüfen, ob das `signer` Feld im JWT-Header den erwarteten ARN der Verified Access-Instanz enthält.

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: Validate the signer
expected_verified_access_instance_arn = 'arn:aws:ec2:region-code:account-id:verified-
access-instance/verified-access-instance-id'

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)
received_verified_access_instance_arn = decoded_json['signer']

assert expected_verified_access_instance_arn == received_verified_access_instance_arn,
 "Invalid Signer"

# Step 2: Get the key id from JWT headers (the kid field)
kid = decoded_json['kid']

# Step 3: 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 4: Get the payload
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES384'])
```

Verifizierte Zugriffsrichtlinien

AWS Verified Access Mithilfe von Richtlinien können Sie Regeln für den Zugriff auf Ihre in gehosteten Anwendungen definieren AWS. Sie sind in Cedar, einer AWS Richtliniensprache, verfasst. Mit Cedar können Sie Richtlinien erstellen, die anhand der Vertrauensdaten bewertet werden, die von den identitäts- oder gerätebasierten Vertrauensanbietern gesendet werden, die Sie für die Verwendung mit Verified Access konfigurieren.

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

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. Alternativ können Sie eine Richtlinie für eine bestehende Gruppe oder einen Endpunkt mit verifiziertem Zugriff hinzufügen oder ändern, nachdem dieser erstellt wurde.

Inhalt

Struktur der Erklärung zur Verified Access-Richtlinie

Die folgende Tabelle zeigt die Struktur einer Richtlinie für verifizierten Zugriff.

Komponente	Syntax
Auswirkung	`permit | forbid`
scope	`(principal, action, resource)`

Komponente	Syntax
Bedingungsklausel	`when { context.policy-reference-name .attribute-name };`

Komponenten der Richtlinie

Eine Richtlinie für verifizierten Zugriff umfasst die folgenden Komponenten:

- Auswirkung — Zugriff entweder `permit` (zulassen) oder `forbid` (verweigern).
- Geltungsbereich — Die Prinzipien, Aktionen und Ressourcen, für die der Effekt gilt. Sie können den Geltungsbereich in Cedar undefiniert lassen, indem Sie keine bestimmten Prinzipale, Aktionen oder Ressourcen angeben. In diesem Fall gilt die Richtlinie für alle möglichen Prinzipale, Aktionen und Ressourcen.
- Bedingungsklausel — Der Kontext, in dem der Effekt gilt.

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

Kommentare

Sie können Kommentare in Ihre AWS Verified Access Richtlinien aufnehmen. Kommentare sind als Zeilen definiert, die mit einem Zeilenumbruchzeichen beginnen `//` und damit enden.

Das folgende Beispiel zeigt Kommentare in einer Richtlinie.

```
// grants access to users in a specific domain using 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)
};
```

Mehrere Klauseln

Mithilfe des && Operators können Sie in einer Richtlinienerklärung mehr als eine Bedingungsklausel verwenden.

```
permit(principal,action,resource)
when{
 context.policy-reference-name.attribute1 &&
 context.policy-reference-name.attribute2
};
```

Weitere Beispiele finden Sie unter Beispielrichtlinien für verifizierten Zugriff.

Reservierte Zeichen

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

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

Integrierte Operatoren für Richtlinien für verifizierten Zugriff

Wenn Sie den Kontext einer AWS Verified Access Richtlinie unter Verwendung verschiedener Bedingungen erstellen, können Sie, wie unter beschriebenStruktur der Erklärung zur Verified Access-Richtlinie, 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, 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).

Operator	Typen und Überladungen	Beschreibung
.existiert ()	Entität → Boolean	Existenz einer Entität.
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.

Verifizierte Bewertung der Zugriffsrichtlinie

Ein Richtliniendokument 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.

AWS Verified Access validiert die Syntax, wenn Sie die Richtlinie erstellen, validiert jedoch nicht die Daten, die Sie in die Bedingungsklausel eingegeben haben.

Verifizierter Kurzschluss der Logik der Zugriffsrichtlinie

Möglicherweise möchten Sie eine AWS Verified Access Richtlinie 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 einen Fehler und bewertet die Richtlinie, um den Zugriff zu verweigern, unabhängig von Ihrer Absicht. Dies würde zum Beispiel zu einer Ablehnung führen, da sie in diesem Kontext `bogus_key` nicht existieren `fake_provider` und 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 „Falsch“ zurückgibt, wird die weitere Auswertung der verketteten Anweisung angehalten, und Cedar gibt beim Versuch, auf ein Element zu verweisen, das nicht existiert, keinen Fehler aus.

```
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 für verifizierten Zugriff

Sie können Richtlinien für verifizierten Zugriff verwenden, um bestimmten Benutzern und Geräten Zugriff auf Ihre Anwendungen zu gewähren.

Beispielrichtlinien

- Beispiel 1: Gewähren Sie einer Gruppe im IAM Identity Center Zugriff
- Beispiel 2: Gewähren Sie Zugriff auf eine Gruppe bei einem Drittanbieter
- Beispiel 3: Zugriff gewähren mit CrowdStrike
- Beispiel 4: Eine bestimmte IP-Adresse zulassen oder verweigern

Beispiel 1: Gewähren Sie einer Gruppe im IAM Identity Center Zugriff

Bei der Verwendung AWS IAM Identity Center ist es besser, Gruppen mit ihren IDs zu bezeichnen. Auf diese Weise können Sie vermeiden, dass gegen eine Richtlinienaussage verstoßen wird, wenn Sie den Namen der Gruppe ändern.

Die folgende Beispielrichtlinie ermöglicht den Zugriff nur Benutzern in der angegebenen Gruppe mit einer verifizierten E-Mail-Adresse. Die Gruppen-ID lautet c242c5b0-6081-1845-6fa8-6e0d9513c107.

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

Die folgende Beispielrichtlinie erlaubt den Zugriff nur, wenn sich der Benutzer in der angegebenen Gruppe befindet, der Benutzer über eine verifizierte E-Mail-Adresse verfügt und die Geräterisikobewertung von Jamf lautetLOW.

```
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"
};
```

Weitere Informationen zu den Vertrauensdaten finden Sie unterthe section called “AWS IAM Identity Center Kontext”.

Beispiel 2: Gewähren Sie Zugriff auf eine Gruppe bei einem Drittanbieter

Die folgende Beispielrichtlinie erlaubt den Zugriff nur, wenn sich der Benutzer in der angegebenen Gruppe befindet, der Benutzer über eine verifizierte E-Mail-Adresse verfügt und der Jamf-Geräterisikowert NIEDRIG ist. Der Name der Gruppe lautet „Finanzen“.

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

Weitere Hinweise zu den Vertrauensdaten finden Sie unterthe section called “Kontext eines Drittanbieters”.

Beispiel 3: Zugriff gewähren mit 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: Eine bestimmte IP-Adresse zulassen oder verweigern

Die folgende Beispielrichtlinie erlaubt Anfragen nur von der angegebenen IP-Adresse.

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

Die folgende Beispielrichtlinie lehnt Anfragen von der angegebenen IP-Adresse ab.

```
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 2: Richtlinien testen und bearbeiten
- Schritt 3: Änderungen überprüfen und anwenden

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: Änderungen überprüfen und anwenden

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.

Connectivity Client für AWS Verified Access

AWS Verified Access stellt den Connectivity Client bereit, sodass Sie Konnektivität zwischen Benutzergeräten und Nicht-HTTP-Anwendungen aktivieren können. Der Client verschlüsselt den Benutzerverkehr sicher, fügt Informationen zur Benutzeridentität und den Gerätekontext hinzu und leitet ihn zur Durchsetzung von Richtlinien an Verified Access weiter. Wenn die Zugriffsrichtlinien den Zugriff zulassen, ist der Benutzer mit der Anwendung verbunden. Der Benutzerzugriff wird kontinuierlich autorisiert, solange der Connectivity Client verbunden ist.

Der Client wird als Systemdienst ausgeführt und ist widerstandsfähig gegen Abstürze. Wenn die Verbindung instabil wird, stellt der Client die Verbindung wieder her.

Der Client verwendet kurzlebige OAuth Zugriffstoken, um den sicheren Tunnel einzurichten. Der Tunnel wird getrennt, wenn sich der Benutzer vom Client abmeldet.

Zugriffs- und Aktualisierungstoken werden lokal auf dem Benutzergerät in einer verschlüsselten SQLite Datenbank gespeichert.

Inhalt

Voraussetzungen

Stellen Sie vor Beginn sicher, dass die folgenden Voraussetzungen erfüllt sind:

- Erstellen Sie eine Instanz mit verifiziertem Zugriff bei einem Vertrauensanbieter.
- Erstellen Sie einen TCP-Endpunkt für Ihre Anwendung.
- Trennen Sie Ihren Computer von allen VPN-Clients, um Routing-Probleme zu vermeiden.

- Aktivieren Sie es IPv6 auf Ihrem Computer. Anweisungen finden Sie in der Dokumentation für das Betriebssystem, das auf Ihrem Computer ausgeführt wird.
- Stellen Sie auf einem Windows-Computer sicher, dass Trusted Platform Module (TPM) unterstützt wird, und installieren Sie die WebView2-Runtime.

Laden Sie den Connectivity Client herunter

Deinstallieren Sie alle früheren Versionen des Clients. Laden Sie den Client herunter, stellen Sie sicher, dass das Installationsprogramm signiert ist, und führen Sie das Installationsprogramm aus. Installieren Sie den Client nicht mit einem unsignierten Installationsprogramm.

- Connectivity Client für Mac mit Apple Silicon Version 1.0.2
- Connectivity Client für Mac mit Intel Version 1.0.2
- Connectivity Client für Windows mit x64 Version 1.0.2

Exportieren der Client-Konfigurationsdatei

Gehen Sie wie folgt vor, um die vom Client benötigten Konfigurationsinformationen aus Ihrer Verified Access-Instanz zu exportieren.

Um die Client-Konfigurationsdatei mit der Konsole zu exportieren

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 Aktionen, Client-Konfigurationsdatei exportieren.

Um die Client-Konfigurationsdatei mit dem zu exportieren AWS CLI

Verwenden Sie den Befehl export-verified-access-instance-client-configuration. Speichern Sie die Ausgabe in einer JSON-Datei. Der Dateiname muss mit dem `ClientConfig-` Präfix beginnen.

Connect zur Anwendung her

Gehen Sie wie folgt vor, um über den Client eine Verbindung zu einer Anwendung herzustellen.

So stellen Sie mithilfe des Clients eine Verbindung zu einer Anwendung her

1. Stellen Sie die Client-Konfigurationsdateien auf den Geräten der Benutzer am folgenden Speicherort bereit:
   - Windows — `C:\ProgramData\Connectivity Client`
   - macOS — `/Library/Application\ Support/Connectivity\ Client`
2. Stellen Sie sicher, dass die Client-Konfigurationsdateien Root (macOS) oder Admin (Windows) gehören.
3. Starten Sie den Connectivity Client.
4. Nachdem der Connectivity Client geladen wurde, wird der Benutzer vom IdP authentifiziert.
5. Nach der Authentifizierung können Benutzer mit dem von Verified Access bereitgestellten DNS-Namen auf die Anwendung zugreifen, indem sie den Client ihrer Wahl verwenden.

Deinstallieren Sie den Client

Wenn Sie den Connectivity Client nicht mehr verwenden, können Sie ihn deinstallieren.

macOS

Version 1.0.1 und höher

Navigieren Sie zum Ordner `/Applications/Connectivity Client` und führen Sie die Datei `Connectivity Client Uninstaller.app` aus.

Version 1.0.0

Laden Sie das `connectivity_client_cleanup.sh` Skript für Mac mit Apple Silicon oder Mac mit Intel herunter, legen Sie Ausführungsberechtigungen für das Skript fest und führen Sie das Skript wie folgt aus.

```
sudo ./connectivity_client_cleanup.sh
```

Windows

Um den Client unter Windows zu deinstallieren, führen Sie das Installationsprogramm aus und wählen Sie Entfernen.

Bewährte Methoden

Beachten Sie die folgenden bewährten Methoden:

- Installieren Sie die neueste Version des Clients.
- Installieren Sie den Client nicht mit einem unsignierten Installationsprogramm.
- Benutzer sollten eine Konfiguration nur verwenden, wenn es sich um eine vertrauenswürdige Konfiguration handelt, die von einem IT-Administrator bereitgestellt wurde. Eine nicht vertrauenswürdige Konfiguration könnte zu einer Phishing-Seite weiterleiten.
- Benutzer sollten sich vom Client abmelden, bevor sie ihre Workstations inaktiv lassen.
- Fügen Sie den `offline_access` Bereich zu Ihrer OIDC-Konfiguration hinzu. Dies ermöglicht Anfragen nach Aktualisierungstoken, die verwendet werden, um mehr Zugriffstoken zu erhalten, ohne dass sich der Benutzer erneut authentifizieren muss.

Fehlerbehebung

Die folgenden Informationen können Ihnen bei der Behebung von Problemen mit dem Client helfen.

Problembereiche

- Bei der Anmeldung wird der Browser nicht geöffnet, um die Authentifizierung durch den IdP abzuschließen
- Nach der Authentifizierung lautet der Client-Status „Nicht verbunden“
- Es kann keine Verbindung mit einem Chrome- oder Edge-Browser hergestellt werden

Bei der Anmeldung wird der Browser nicht geöffnet, um die Authentifizierung durch den IdP abzuschließen

Mögliche Ursache: Die Konfigurationsdatei fehlt oder ist falsch formatiert.

Lösung: Wenden Sie sich an Ihren Systemadministrator und fordern Sie eine aktualisierte Konfigurationsdatei an.

Nach der Authentifizierung lautet der Client-Status „Nicht verbunden“

Mögliche Ursache: Ausführung anderer VPN-Software wie AWS Client VPN Cisco AnyConnect oder OpenVPN Connect.

Lösung: Trennen Sie die Verbindung zu jeder anderen VPN-Software. Wenn Sie immer noch keine Verbindung herstellen können, erstellen Sie einen Diagnosebericht und teilen Sie ihn Ihrem Systemadministrator mit.

Mögliche Ursache: Auf Windows-Plattformen verwendet der Client HTTP auf Port 80 für die Kommunikation auf der Steuerungsebene. Eine Firewallregel, die den TCP-Port 80 blockiert, verhindert die Kommunikation auf der Kontrollebene.

Lösung: Suchen Sie in den Windows-Firewallregeln nach einer expliziten Regel für ausgehenden Datenverkehr, die TCP auf Port 80 blockiert, und deaktivieren Sie sie.

Es kann keine Verbindung mit einem Chrome- oder Edge-Browser hergestellt werden

Mögliche Ursache: Wenn Sie über einen Chrome- oder Edge-Browser eine Verbindung zu einer Webanwendung herstellen, kann der Browser den IPv6 Domainnamen nicht auflösen.

Lösung: Kontakt AWS -Support.

Versionshistorie

Die folgende Tabelle enthält den Versionsverlauf des Clients.

Version	Änderungen	Herunterladen	Datum
1.0.2	macOS • Fehlerkorrekturen und Stabilitätsverbesserungen • Verbesserungen der Benutzeroberfläche Windows • Fehlerkorrekturen und Stabilitätsverbesserungen • Verbesserungen der Benutzeroberfläche	• Mac mit Apple Silicon • Mac mit Intel • Windows mit x64	9. Juni 2025

Version	Änderungen	Herunterladen	Datum
1.0.1	macOS • Verbesserungen der Stabilität • Deinstallationsprogramm Windows • Verbesserungen der Stabilität	• Mac mit Apple Silicon • Mac mit Intel • Windows mit x64	5. Februar 2025
1.0.0	Öffentliche Vorschau	• Mac mit Apple Silicon • Mac mit Intel • Windows mit x64	01. Dezember 2024

Sicherheit bei verifiziertem Zugriff

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für AWS Verified Access gelten, finden Sie unter AWS Services im Umfang nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. 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 bei Verified Access

Das AWS Modell der gilt für den Datenschutz in AWS Verified Access. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle

Systeme laufen AWS Cloud. 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, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. 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, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen 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 AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

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 auf andere Weise AWS-Services über die Konsole, API oder arbeiten. AWS CLI AWS SDKs 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, die von Endbenutzern zu Verified Access-Endpunkten über das Internet übertragen werden, mit Transport Layer Security (TLS) 1.2 oder höher.

Datenschutz für den Datenverkehr zwischen Netzwerken

Sie können Verified Access konfigurieren, um den Zugriff auf bestimmte Ressourcen in Ihrer VPC einzuschränken. Bei der benutzerbasierten 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 verifizierten Zugriff AWS

AWS Verified Access verschlüsselt standardmäßig ruhende Daten mithilfe AWS eigener KMS-Schlüssel. Wenn die Verschlüsselung ruhender Daten standardmäßig erfolgt, trägt dies dazu bei, den betrieblichen Aufwand und die Komplexität zu reduzieren, die mit dem Schutz vertraulicher Daten verbunden sind. Gleichzeitig können Sie damit sichere Anwendungen erstellen, die strenge Verschlüsselungsvorschriften und gesetzliche Auflagen erfüllen. In den folgenden Abschnitten wird detailliert beschrieben, wie Verified Access KMS-Schlüssel für die Verschlüsselung inaktiver Daten verwendet.

Inhalt

Verifizierter Zugriff und KMS-Schlüssel

AWS eigene Schlüssel

Verified Access verwendet KMS-Schlüssel, um personenbezogene Daten (PII) automatisch zu verschlüsseln. Dies geschieht standardmäßig, und Sie können die Verwendung der AWS-eigenen

Schlüssel nicht selbst einsehen, 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 AWS eigenen Verschlüsselungsschlüsseln hinzufügen, indem Sie bei der Erstellung Ihrer verifizierten Zugriffsressourcen 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 der vorhandenen Standardverschlüsselung eine zweite Verschlüsselungsebene 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 Service Entwicklerhandbuch.

> **Note**
>
> Verified Access ermöglicht automatisch die Verschlüsselung im Ruhezustand mithilfe AWS eigener Schlüssel, um personenbezogene Daten kostenlos zu schützen.
> Es AWS KMS fallen jedoch Gebühren an, wenn Sie einen vom Kunden verwalteten Schlüssel verwenden. Weitere Informationen zur Preisgestaltung finden Sie unter AWS Key Management Service Preisgestaltung.

Persönlich identifizierbare Informationen

In der folgenden Tabelle werden die von Verified Access verwendeten personenbezogenen Daten (PII) und deren Verschlüsselung zusammengefasst.

Datentyp	AWS Verschlüsselung mit eigenem Schlüssel	Vom Kunden verwaltete Schlüsselverschlüsselung (optional)
`Trust provider (user-type)` Vertrauensanbieter vom Typ Benutzer enthalten OIDC-Optionen wie AuthorizationEndpoint,, UserInfoEndpoint ClientId, usw. ClientSecret, die als personenbezogene Daten betrachtet werden.	Aktiviert	Aktiviert
`Trust provider (device-type)` Vertrauensanbieter vom Typ Gerät enthalten eine TenantId, die als PII betrachtet wird.	Aktiviert	Aktiviert
`Group policy` Wird bei der Erstellung oder Änderung der Verified Access-Gruppe bereitgestellt. Enthält Regeln für die Autorisierung von Zugriffsanfragen. Kann personenbezogene Daten wie Benutzername und E-Mail-Adresse usw. enthalten.	Aktiviert	Aktiviert
`Endpoint policy`	Aktiviert	Aktiviert

Datentyp	AWS Verschlüsselung mit eigenem Schlüssel	Vom Kunden verwaltete Schlüsselverschlüsselung (optional)
Wird bei der Erstellung oder Änderung des Verified Access-Endpunkts bereitgestellt. Enthält Regeln für die Autorisierung von Zugriffsanfragen. Kann personenbezogene Daten wie Benutzername und E-Mail-Adresse usw. enthalten.		

So verwendet AWS Verified Access Zuschüsse in AWS KMS

Für Verified Access ist eine Genehmigung erforderlich, um Ihren vom Kunden verwalteten Schlüssel verwenden zu können.

Wenn Sie Ressourcen mit verifiziertem Zugriff erstellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt sind, erstellt Verified Access in Ihrem Namen einen Zuschuss, indem es eine CreateGrantAnfrage an sendet AWS KMS. Grants in AWS KMS werden verwendet, um Verified Access den Zugriff auf einen vom Kunden verwalteten Schlüssel in Ihrem Konto zu gewähren.

Verified Access setzt voraus, dass der Zuschuss Ihren vom Kunden verwalteten Schlüssel für die folgenden internen Vorgänge verwendet:

- Senden Sie Entschlüsselungsanfragen an AWS KMS , um die verschlüsselten Datenschlüssel zu entschlüsseln, sodass sie zur Entschlüsselung Ihrer Daten verwendet werden können.
- Senden Sie RetireGrantAnfragen an, um einen AWS KMS Zuschuss 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 wurden, was sich auf Vorgänge auswirkt, die von diesen Daten abhängig sind.

Verwenden von vom Kunden verwalteten Schlüsseln mit verifiziertem Zugriff

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

Die wichtigsten Richtlinien

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 Wichtige Richtlinien im AWS Key Management Service Entwicklerhandbuch.

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren verifizierten Zugriffsressourcen verwenden zu können, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zugelassen sein:

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

  Dadurch 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 Principal ein, der in den Ruhestand geht, `RetireGrant` damit der Dienst
- `kms:DescribeKey`— Stellt dem Kunden verwaltete Schlüsseldetails zur Verfügung, damit Verified Access den Schlüssel überprüfen kann.
- `kms:GenerateDataKey`— Ermöglicht Verified Access, den Schlüssel zum Verschlüsseln von Daten zu verwenden.
- `kms:Decrypt`— Erlaubt 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 finden Sie unter Erstellen einer Schlüsselrichtlinie und Problembehandlung beim Schlüsselzugriff im AWS Key Management Service Entwicklerhandbuch.

Angabe 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:

- Gruppe „Verifizierter Zugriff“
- Verifizierter Zugriffsendpunkt
- Verifizierter Access-Vertrauensanbieter

Wenn Sie eine dieser Ressourcen mit dem erstellen AWS 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 ist auch möglich, wenn Sie eine vorhandene Ressource ändern, oder indem Sie den verwenden AWS CLI.

> **Note**
>
> Wenn der vom Kunden verwaltete Schlüssel, mit dem eine zusätzliche Verschlüsselung für eine der oben genannten Ressourcen hinzugefügt wurde, verloren geht, sind die Konfigurationswerte für die Ressourcen nicht mehr zugänglich. Die Ressourcen können jedoch geändert werden, indem das AWS Management Console oder verwendet wird AWS CLI, um einen neuen, vom Kunden verwalteten Schlüssel anzuwenden und die Konfigurationswerte zurückzusetzen.

AWS Verschlüsselungskontext für verifizierten Zugriff

Ein Verschlüsselungskontext ist ein optionaler Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten. AWS KMS verwendet den Verschlüsselungskontext als zusätzliche authentifizierte Daten, um die authentifizierte Verschlüsselung zu unterstützen. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zum Verschlüsseln von Daten einbeziehen, wird der Verschlüsselungskontext AWS KMS an die verschlüsselten Daten gebunden. Zur Entschlüsselung von Daten müssen Sie denselben Verschlüsselungskontext in der Anfrage übergeben.

AWS Verifizierter Access-Verschlüsselungskontext

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

Vertrauensanbieter mit verifiziertem Zugriff

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

Gruppe „Verifizierter Zugriff“

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

Verifizierter Zugriffsendpunkt

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

Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS verifizierten Zugriff

Wenn Sie einen vom Kunden verwalteten KMS-Schlüssel mit Ihren AWS Verified Access-Ressourcen verwenden, können Sie AWS CloudTraildamit Anfragen verfolgen, an die Verified Access sendet AWS KMS.

Bei den folgenden Beispielen handelt es sich um AWS CloudTrail Ereignisse für`CreateGrant`,`RetireGrant`, und `Decrypt` `DescribeKey``GenerateDataKey`, mit denen KMS-Vorgänge überwacht werden, die von Verified Access aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten KMS-Schlüssel verschlüsselt wurden:

CreateGrant

Wenn Sie einen vom Kunden verwalteten Schlüssel zur Verschlüsselung Ihrer Ressourcen verwenden, sendet Verified Access in Ihrem Namen eine `CreateGrant` Anfrage, um auf den Schlüssel in Ihrem AWS Konto zuzugreifen. Die Gewährung, die Verified Access gewährt, 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 den `RetireGrant` Vorgang, um einen Zuschuss 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 den `Decrypt` Vorgang auf, um mithilfe des gespeicherten verschlüsselten Datenschlüssels auf die verschlüsselten Daten zuzugreifen.

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 den `DescribeKey` Vorgang, um zu überprüfen, ob der vom Kunden verwaltete 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 Verified Access

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert

(angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen mit verifiziertem Zugriff zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt davon ab, welche Arbeit Sie mit 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 Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie unter Verifizierter Zugriff nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unterFehlerbehebung bei 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. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer 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 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

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

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

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter AWS Signature Version 4 für API-Anforderungen im IAM-Benutzerhandbuch.

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

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-

Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, 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 sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, 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 in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten 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 innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. 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 eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung 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 Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln. Sie können eine Rolle übernehmen, 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 Methoden für die Übernahme einer Rolle 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 (Verbund) 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. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. 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 aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. 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.
  - Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwenden einer

IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden 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 angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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 Rolleninformationen von der AWS Management Console AWS CLI, der oder der 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 Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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 System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter Auswählen 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. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

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

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete 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. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter Resource Control Policies (RCPs) 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 darüber, wie AWS bestimmt wird,

ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

So funktioniert 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-Feature	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
Prinzipalberechtigungen	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 Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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

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. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

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 unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos 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 Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für verifizierten Zugriff

Unterstützt Richtlinienaktionen: Ja

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

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 der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang 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 definierte Aktionen EC2 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

Richtlinienressourcen für verifizierten Zugriff

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und 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 deren Eigenschaften ARNs finden Sie unter Von Amazon definierte Ressourcen EC2 in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon definierte Aktionen EC2.

Beispiele für identitätsbasierte Richtlinien mit verifiziertem Zugriff finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access

Bedingungsschlüssel für Richtlinien für verifizierten Zugriff

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. 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, AWS wertet die Bedingung mithilfe 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 dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

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

Beispiele für identitätsbasierte Richtlinien mit verifiziertem Zugriff finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Verified Access

ACLs unter Verifizierter Zugriff

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten 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 an viele AWS Ressourcen anhängen. 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 Bedingung `aws:TagKeys` 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 Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit verifiziertem Zugriff

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. 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 von einer Benutzerrolle zu einer IAM-Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu 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 verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. 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 aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. 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 dienstverknüpfte Rollen: Ja

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

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

Beispiele für identitätsbasierte Richtlinien für Verified Access

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Ressourcen mit verifiziertem Zugriff zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder 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 (Konsole) im IAM-Benutzerhandbuch.

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

Themen

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 verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen 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 Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. 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 mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, 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 Sicherer API-Zugriff mit MFA 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 Verified Access-Instanzen

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` ist es 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 umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-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 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 Ressourcen mit verifiziertem Zugriff 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.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. 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 ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. 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.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Ressourcen mit verifiziertem Zugriff 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. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, 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 unter. So funktioniert Verified Access mit IAM
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- 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 Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwenden Sie serviceverknüpfte Rollen für verifizierten Zugriff

AWS Verified Access verwendet eine mit dem Dienst verknüpfte IAM-Rolle. Dabei handelt es sich um eine Art von IAM-Rolle, die direkt mit einem Dienst verknüpft ist. AWS Die dienstbezogenen Rollen für Verified Access werden von Verified Access definiert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS-Services in Ihrem Namen anzurufen.

Eine dienstbezogene Rolle erleichtert die Einrichtung von Verified Access, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Verified Access definiert die Berechtigungen seiner dienstbezogenen Rollen, und sofern nicht anders definiert, kann nur Verified Access seine Rollen übernehmen. Zu den definierten Berechtigungen gehören die Vertrauensrichtlinie und die

Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keiner anderen IAM-Entität zugeordnet werden.

Mit dem Dienst verknüpfte Rollenberechtigungen für verifizierten Zugriff

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

Die mit dem Dienst verknüpfte AWSServiceRoleForVPCVerifiedAccess-Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `verified-access.amazonaws.com`

Die genannte Rollenberechtigungsrichtlinie ermöglicht es Verified Access AWSVPCVerifiedAccessServiceRolePolicy, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion `ec2:CreateNetworkInterface` für alle Subnetze und Sicherheitsgruppen sowie für alle Netzwerkschnittstellen mit dem Tag `VerifiedAccessManaged=true`
- Aktion `ec2:CreateTags` für alle 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 Referenzhandbuch für AWS verwaltete Richtlinien einsehen; siehe AWSVPCVerifiedAccessServiceRolePolicy.

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

Erstellen Sie eine dienstbezogene Rolle für Verified Access

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie die AWS Management Console, oder die AWS API aufrufen CreateVerifiedAccessEndpoint AWS CLI, erstellt Verified Access die dienstbezogene 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 aufrufen, erstellt Verified Access die serviceverknüpfte Rolle erneut für Sie.

Bearbeiten Sie eine dienstverknüpfte Rolle für Verified Access

Mit Verified Access können Sie die serviceverknüpfte AWSServiceRoleForVPCVerifiedAccess-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 im IAM-Benutzerhandbuch unter Bearbeiten einer Beschreibung einer dienstbezogenen Rolle.

Löschen Sie eine dienstverknüpfte Rolle für Verified Access

Sie müssen die AWSServiceRoleForVPCVerifiedAccess-Rolle nicht manuell löschen. Wenn Sie die AWS Management Console, oder die AWS API aufrufen DeleteVerifiedAccessEndpoint AWS CLI, bereinigt Verified Access die Ressourcen und löscht die dienstbezogene Rolle für Sie.

So löschen Sie die serviceverknüpfte Rolle mit IAM

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

Unterstützte Regionen für dienstverknüpfte Rollen mit verifiziertem Zugriff

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

AWS verwaltete Richtlinien für verifizierten Zugriff

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Denken Sie daran, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie für alle

AWS Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service 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: AWSVPCVerified AccessServiceRolePolicy

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 ConsoleAWSVPCVerifiedAccessServiceRolePolicyRichtlinie 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 nachzuverfolgen. 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 seine 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 aktualisi	31. Mai 2023

Änderung	Beschreibung	Datum
	ert, um der `ec2:CreateNetworkInterface` Berechtigung eine Sicherheitsgruppenressource hinzuzufügen.	
AWSVPCVerifiedAccessServiceRolePolicy – Neue Richtlinie	Verified Access hat eine neue Richtlinie hinzugefügt, mit der Ressourcen in Ihrem Konto bereitgestellt werden können, die für die Nutzung des Dienstes erforderlich sind.	29. November 2022
Verified Access hat mit der Nachverfolgung von Änderungen begonnen	Verified Access hat damit begonnen, Änderungen an den AWS verwalteten Richtlinien nachzuverfolgen.	29. November 2022

Konformitätsprüfung für verifizierten Zugriff

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

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

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

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services 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.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz bei verifiziertem Zugriff

Die AWS globale Infrastruktur basiert auf AWS-Regionen Availability Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander 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 zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale 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 Endpunkt vom Typ Verified Access 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 AWS Verified Access

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS Verified Access. AWS bietet die folgenden Überwachungstools, um Verified Access zu überwachen, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Zugriffsprotokolle — Erfassen Sie detaillierte Informationen über Anfragen zum 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 Ihnen oder in Ihrem Namen getätigt wurden, AWS-Konto und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Aufrufe erfolgten. Weitere Informationen finden Sie unter the section called “CloudTrail protokolliert”.

Protokolle für verifizierten Zugriff

Nach der AWS Verified Access Auswertung jeder Zugriffsanfrage werden alle Zugriffsversuche protokolliert. Auf diese Weise erhalten Sie einen zentralen Überblick über den Anwendungszugriff und können schnell auf Sicherheitsvorfälle und Prüfanfragen reagieren. Verified Access unterstützt das Protokollierungsformat Open Cybersecurity Schema Framework (OCSF).

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

- Amazon CloudWatch Logs-Protokollgruppen
- Amazon-S3-Buckets
- Amazon Data Firehose-Lieferstreams

Inhalt

Verifizierte Versionen der Zugriffsprotokollierung

Standardmäßig verwendet das Protokollierungssystem für verifizierten Zugriff das Open Cybersecurity Schema Framework (OCSF) Version 0.1. Beispielprotokolle, die Version 0.1 verwenden, finden Sie unterOCSF-Protokollbeispiele der Version 0.1 für verifizierten Zugriff.

Die neueste Logging-Version ist mit der OCSF-Version 1.0.0-rc.2 kompatibel. Weitere Informationen zum Schema finden Sie unter OCSF-Schema. Beispielprotokolle, die Version 1.0.0-rc.2 verwenden, finden Sie unter. OCSF-Protokollbeispiele der Version 1.0.0-rc.2 für verifizierten Zugriff

Beachten Sie, dass Sie OCSF Version 0.1 nicht verwenden können, wenn der Verified Access-Endpunkt das TCP-Protokoll verwendet.

Um die Logging-Version mithilfe der Konsole zu aktualisieren

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 entsprechende Verified Access-Instanz aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Verified Access-Instanzprotokollierung die Option Konfiguration der Verified Access-Instanzprotokollierung ändern aus.
5. Wählen Sie ocsf-1.0.0-rc.2 aus der Dropdownliste Protokollversion aktualisieren aus.
6. Wählen Sie Konfiguration für die Protokollierung der Instanz mit verifiziertem Zugriff ändern aus.

Um die Logging-Version mit dem zu aktualisieren AWS CLI

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

Verifizierte Zugriffsprotokollierungsberechtigungen

Der IAM-Prinzipal, der zur Konfiguration des Protokollierungsziels verwendet wird, benötigt bestimmte Berechtigungen, damit die Protokollierung ordnungsgemäß funktioniert. In den folgenden Abschnitten werden die Berechtigungen aufgeführt, die für jedes Protokollierungsziel erforderlich sind.

Für die Lieferung an CloudWatch Logs:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`auf der Verified Access-Instanz
- `logs:CreateLogDelivery`,`logs:DeleteLogDelivery`, `logs:GetLogDeliverylogs:ListLogDeliveries`, und `logs:UpdateLogDelivery` auf allen Ressourcen
- `logs:DescribeLogGroupslogs:DescribeResourcePolicies`, und in `logs:PutResourcePolicy` der Zielprotokollgruppe

Für die Lieferung an Amazon S3:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`auf der Verified Access-Instance
- `logs:CreateLogDelivery`,`logs:DeleteLogDelivery`, `logs:GetLogDeliverylogs:ListLogDeliveries`, und `logs:UpdateLogDelivery` auf allen Ressourcen
- `s3:GetBucketPolicy`und `s3:PutBucketPolicy` auf dem Ziel-Bucket

Für die Lieferung an Firehose:

- `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`auf der Verified Access-Instanz
- `firehose:TagDeliveryStream`auf allen Ressourcen
- `iam:CreateServiceLinkedRole`auf allen Ressourcen
- `logs:CreateLogDelivery`,`logs:DeleteLogDelivery`, `logs:GetLogDeliverylogs:ListLogDeliveries`, und `logs:UpdateLogDelivery` auf allen Ressourcen

Aktivieren oder deaktivieren Sie Protokolle für verifizierten Zugriff

Sie können die Verfahren in diesem Abschnitt verwenden, um die Protokollierung zu aktivieren oder zu deaktivieren. Wenn Sie die Protokollierung aktivieren, müssen Sie ein Ziel für die zu sendenden Protokolle konfigurieren. Der IAM-Prinzipal, der zur Konfiguration des Protokollierungsziels verwendet wird, benötigt bestimmte Berechtigungen, damit die Protokollierung ordnungsgemäß funktioniert. Die erforderlichen IAM-Berechtigungen für jedes Protokollierungsziel finden Sie im Verifizierte Zugriffsprotokollierungsberechtigungen Abschnitt.

Inhalt

Aktivieren der Zugriffsprotokolle

Um Verified Access-Logs zu aktivieren

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 auf der Registerkarte Konfiguration der Verified Access-Instanzprotokollierung die Option Konfiguration der Verified Access-Instanz ändern aus.
5. (Optional) Gehen Sie wie folgt vor, um Vertrauensdaten, die von Vertrauensanbietern gesendet wurden, in die Protokolle aufzunehmen:

   a. Wählen Sie ocsf-1.0.0-rc.2 aus der Dropdownliste Protokollversion aktualisieren aus.

   b. Wählen Sie Vertrauenskontext einbeziehen.
6. Führen Sie eine der folgenden Aktionen aus:
   - Aktivieren Sie Deliver to Amazon CloudWatch Logs. Wählen Sie die Zielprotokollgruppe aus.
   - Aktivieren Sie Deliver to Amazon S3. Geben Sie den Namen, den Besitzer und das Präfix des Ziel-Buckets ein.
   - Aktiviere „An Firehose liefern“. Wählen Sie den Ziel-Lieferstream aus.
7. Wählen Sie „Konfiguration für die Protokollierung der Verified Access-Instance ändern“.

Um Verified Access-Logs zu aktivieren, verwenden Sie AWS CLI

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

Deaktivieren der Zugriffsprotokolle

Sie können die Zugriffsprotokolle für Ihre Verified Access-Instanz jederzeit deaktivieren. Nachdem Sie die Zugriffsprotokolle deaktiviert haben, verbleiben Ihre Protokolldaten in Ihrem Protokollziel, bis Sie sie löschen.

Um Verified Access-Logs deaktivieren

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 auf der Registerkarte Konfiguration der Verified Access-Instanzprotokollierung die Option Konfiguration der Verified Access-Instanz ändern aus.
5. Schalten Sie die Protokollzustellung aus.
6. Wählen Sie Konfiguration für die Protokollierung der Instanz mit verifiziertem Zugriff ändern aus.

Um Verified Access-Logs zu deaktivieren, verwenden Sie AWS CLI

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

Aktivieren oder deaktivieren Sie den Vertrauenskontext Verified Access

Der von Ihrem Vertrauensanbieter gesendete Vertrauenskontext kann optional für die Aufnahme in Ihre Verified Access-Logs aktiviert werden. Dies kann nützlich sein, wenn Sie Richtlinien definieren, die den Zugriff auf Ihre Anwendungen zulassen oder verweigern. Nachdem Sie es aktiviert haben, befindet sich der Vertrauenskontext im Protokoll unter dem `data` Feld. Wenn der Vertrauenskontext deaktiviert ist, ist das `data` Feld auf gesetzt`null`. Gehen Sie wie folgt vor, um Verified Access so zu konfigurieren, dass der Vertrauenskontext in die Protokolle aufgenommen wird.

> **Note**
>
> Um den Vertrauenskontext in Ihre Verified Access-Protokolle aufzunehmen, ist ein Upgrade auf die neueste Protokollierungsversion erforderlich`ocsf-1.0.0-rc.2`. Beim folgenden Verfahren wird davon ausgegangen, dass Sie die Protokollierung bereits aktiviert haben. Falls das nicht zutrifft, finden Sie Aktivieren der Zugriffsprotokolle das vollständige Verfahren unter.

Inhalt

Aktivieren Sie den Vertrauenskontext

Um mithilfe der Konsole den Vertrauenskontext in die Verified Access-Logs aufzunehmen

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 entsprechende Verified Access-Instanz aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Verified Access-Instanzprotokollierung die Option Konfiguration der Verified Access-Instanzprotokollierung ändern aus.
5. Wählen Sie ocsf-1.0.0-rc.2 aus der Dropdownliste Protokollversion aktualisieren aus.
6. Aktivieren Sie die Option Vertrauenskontext einbeziehen.
7. Wählen Sie Konfiguration für die Protokollierung der Instanz „Verified Access“ ändern aus.

Um den Vertrauenskontext in die Verified Access-Logs aufzunehmen, verwenden Sie AWS CLI

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

Deaktivieren Sie den Vertrauenskontext

Wenn Sie den Vertrauenskontext nicht mehr in die Protokolle aufnehmen möchten, können Sie ihn wie folgt entfernen.

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

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 entsprechende Verified Access-Instanz aus.
4. Wählen Sie auf der Registerkarte Konfiguration der Verified Access-Instanzprotokollierung die Option Konfiguration der Verified Access-Instanzprotokollierung ändern aus.
5. Deaktivieren Sie die Option Vertrauenskontext einbeziehen.
6. Wählen Sie Konfiguration für die Protokollierung der Instanz „Verified Access“ ändern aus.

Um den Vertrauenskontext aus den Verified Access-Protokollen zu entfernen, verwenden Sie AWS CLI

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

OCSF-Protokollbeispiele der Version 0.1 für verifizierten Zugriff

Im Folgenden finden Sie Beispielprotokolle, die OCSF Version 0.1 verwenden.

Beispiele

- Zugriff mit OIDC gewährt
- Mit OIDC und JAMF gewährter Zugriff
- Zugriff gewährt mit OIDC und CrowdStrike
- Der Zugriff wurde aufgrund eines fehlenden Cookies verweigert
- Der Zugriff wurde per Richtlinie verweigert
- Unbekannter Protokolleintrag

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
}
```

Mit OIDC und JAMF gewährter Zugriff

In diesem Beispielprotokolleintrag ermöglicht Verified Access den Zugriff auf einen Endpunkt sowohl bei OIDC- als auch bei 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 bei OIDC- CrowdStrike als auch bei Device Trust Providern.

```
{
    "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
}
```

Der Zugriff wurde aufgrund eines fehlenden Cookies verweigert

In diesem Beispielprotokolleintrag verweigert Verified Access den Zugriff aufgrund eines fehlenden Authentifizierungs-Cookies.

```
{
    "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
}
```

Der Zugriff wurde per Richtlinie verweigert

In diesem Beispielprotokolleintrag lehnt Verified Access eine authentifizierte Anfrage ab, da die Anfrage gemäß den Zugriffsrichtlinien nicht zulässig ist.

```
{
    "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 Beispiel kann Verified Access keinen vollständigen Protokolleintrag generieren und gibt daher einen unbekannten Protokolleintrag aus. Dadurch wird sichergestellt, dass jede Anfrage im Zugriffsprotokoll erscheint.

```
{
    "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
}
```


OCSF-Protokollbeispiele der Version 1.0.0-rc.2 für verifizierten Zugriff

Im Folgenden finden Sie Beispielprotokolle, die OCSF Version 1.0.0-rc.2 verwenden.

Beispiele

- Zugriff gewährt, einschließlich Vertrauenskontext
- Zugriff gewährt, obwohl der Vertrauenskontext weggelassen wurde
- Weisen Sie dem Netzwerk-CIDR-Endpunkt Rechte zu

Zugriff gewährt, einschließlich Vertrauenskontext


```
{
    "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",
                "additional_user_context": {
                    "aud": "xxx",
                    "exp": 1000000000,
                    "groups": [
                        "group-id-1",
                        "group-id-2"
                    ],
                    "iat": 1000000000,
                    "iss": "https://oidc-tp.com/",
                    "sub": "xyzsubject",
                    "ver": "1.0"
                }
            },
            "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 gewährt, obwohl der Vertrauenskontext weggelassen wurde

```
{
    "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
}
```


Weisen Sie dem Netzwerk-CIDR-Endpunkt Rechte zu


```
{
    "activity_id": "1",
    "activity_name": "Assign Privileges",
    "category_name": "Audit Activity",
    "category_uid": "3",
```

```
    "class_name": "Authorization",
    "class_uid": "3003",
    "data": {
        "endpoint_type": "cidr",
        "protocol": "tcp",
        "access_path": "public",
        "idp": {
            "name": "my-oidc-instance",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "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",
                "additional_user_context": {
                    "aud": "xxx",
                    "exp": 1000000000,
                    "groups": [
                        "group-id-1",
                        "group-id-2"
                    ],
                    "iat": 1000000000,
                    "iss": "https://oidc-tp.com/",
                    "sub": "xyzsubject",
                    "ver": "1.0"
                }
            },
```

```
            "tcp_flow": {
                "destination_ip": "10.0.0.1",
                "destination_port": 22,
                "client_ip": "10.2.7.68"
            }
        }
    },
    "device": {
        "ip":  "10.2.7.68",
        "port": 1002,
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "metadata": {
        "uid": "",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "severity": "Informational",
    "severity_id": "1",
    "start_time": "1668580194340",
    "status_code": "200",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300301",
    "type_name": "Authorization: Assign Privileges",
    "count": 1,
    "dst_endpoint": {
        "ip": "107.22.231.155",
        "port": 22
    },
    "privileges": [
        "vae-12345cbce2EXAMPLE"
    ],
    "user": {
        "email_addr": "johndoe-user@test.com",
        "uid": "johndoe-user",
```

```
        "uuid": "9bcce02a-fc15-4091-a0b7-874d157c67b8"
    }
}
```

API-Aufrufe für verifizierten Zugriff protokollieren mit AWS CloudTrail

AWS Verified Access ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service in Verified Access ausgeführten Aktionen bereitstellt. CloudTrail erfasst API-Aufrufe für Verified Access als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Verified Access-Konsole und Codeaufrufen für die Verified Access-API-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Verified Access gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

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 Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto , wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem. AWS-Region Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Arbeiten mit dem CloudTrail Ereignisverlauf. Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder CloudTrailLake-Event-Datenspeicher.

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS Management Console sind regionsübergreifend. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten AWS-Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere Informationen zu Trails finden Sie unter Erstellen eines Trails für Ihr AWS-Konto und Erstellen eines Trails für eine Organisation im AWS CloudTrail -Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise. Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3 – Preise.

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format. ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von erweiterten Ereignisselektoren auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter Arbeiten mit AWS CloudTrail Lake im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die Preisoption aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise.

Verifizierte Access Management-Ereignisse

Verwaltungsereignisse enthalten Informationen zu Verwaltungsvorgängen, die an Ressourcen in Ihrem ausgeführt werden AWS-Konto. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail Protokolliert standardmäßig Verwaltungsereignisse.

Verified Access protokolliert Kontrollplanvorgänge als Verwaltungsereignisse. Eine Liste finden Sie in der Amazon EC2 API-Referenz.

Beispiele für Ereignisse mit verifiziertem Zugriff

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das die `CreateVerifiedAccessInstance` Aktion demonstriert.


```
{
    "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"
}
```


Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter CloudTrailDatensatzinhalte.

Kontingente für AWS Verified Access

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

AWS-Konto Kontingente 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

Für HTTP-Header gilt die folgende Größenbeschränkung.

Name	Standard	Anpassbar
Anforderungszeile	16 K	Nein
Einzelner Header	16 K	Nein
Gesamter Antwort-Header	32 K	Nein
Gesamter Anfrage-Header	64 K	Nein

HTTP-Verkehr

Das Zeitlimit für den Leerlauf der Verbindung beträgt 60 Sekunden. Wenn eine Anwendung länger als 60 Sekunden benötigt, um auf eine HTTP-Anfrage zu antworten, erhält der Client einen HTTP 504-Gateway-Timeout-Fehler. Wenn Verified Access-Logs aktiviert sind, protokollieren wir alle HTTP 504-Fehler.

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

IAM Identity Center

Verified Access kann Benutzern in IAM Identity Center, die bis zu 1.000 Gruppen zugewiesen sind, Zugriff gewähren.

Dokumentenverlauf für das Verified Access-Benutzerhandbuch

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

Änderung	Beschreibung	Datum
Support für Zugriffstoken im Vertrauenskontext	Aktualisierung, `additional_user_context` um OIDC-Benutzeransprüche hinzuzufügen.	24. Februar 2025
Support für Ressourcen über Nicht-HTTP-Protokolle	Freigabe des Zugriffs auf Ressourcen über Nicht-HTTP-Protokolle.	5. Februar 2025
Vorschau-Version	Vorschauversion des Zugriffs auf Ressourcen über Nicht-HTTP-Protokolle.	01. Dezember 2024
AWS Die verwaltete Richtlinie wurde aktualisiert	Die AWS verwaltete IAM-Richtlinie für verifizierten Zugriff wurde aktualisiert.	17. November 2023
Datenverschlüsselung im Ruhezustand	AWS Verified 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

AWS Die 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 WAF Integration.	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.