ユーザーガイド

AWS Verified Access

AWS Verified Access: ユーザーガイド

Table of Contents

とは AWS Verified Access

を使用すると AWS Verified Access、仮想プライベートネットワーク () を使用することなく、アプリケーションへの安全なアクセスを提供できますVPN。Verified Access は各アプリケーションリクエストを評価し、指定されたセキュリティ要件を満たす場合にのみユーザーが各アプリケーションにアクセスできるようにサポートします。

Verified Access の利点

- セキュリティ状態の向上 — 従来のセキュリティモデルでは、アクセスを一度評価すると、すべてのアプリケーションへのアクセス権がユーザーに付与されます。Verified Access では、各アプリケーションのアクセスリクエストがリアルタイムで評価されます。これにより、脅威アクターがあるアプリケーションから別のアプリケーションに移動することが困難になります。
- セキュリティサービスとの統合 – Verified Access は、 とサードパーティーサービスの両方を含む ID AWS およびデバイス管理サービスと統合されます。Verified Access は、これらのサービスのデータを使用して、一連のセキュリティ要件に照らしてユーザーとデバイスの信頼性を検証し、ユーザーがアプリケーションに対するアクセス権を所有すべきかどうかを判断します。
- ユーザーエクスペリエンスの向上 – Verified Access では、ユーザーが を使用してアプリケーションVPNにアクセスする必要がなくなります。これにより、 VPN関連の問題から発生するサポートケースの数を減らすことができます。
- トラブルシューティングと監査の簡素化 — Verified Access はすべてのアクセス試行を記録し、アプリケーションへのアクセスを一元的に把握できるため、セキュリティインシデントや監査請求に迅速に対応できます。

Verified Access へのアクセス

次のいずれかのインターフェイスを使用して Verified Access を操作できます。

- AWS Management Console – Verified Access リソースの作成と管理に使用できるウェブインターフェイスを提供します。にサインイン AWS Management Console し、 で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
- AWS Command Line Interface （ AWS CLI ） – AWS のサービスを含む幅広い セットのコマンドを提供します AWS Verified Access。 AWS CLI は、Windows、macOSでサポートされています。を取得するには AWS CLI、「」を参照してくださいAWS Command Line Interface。

- AWS SDKs – 言語固有の を提供しますAPIs。 AWS SDKs は、署名の計算やリクエストの再試行やエラーの処理など、接続の詳細の多くを処理します。詳細については、「」を参照してくださいAWS SDKs。
- クエリ API — HTTPSリクエストを使用して呼び出す低レベルのAPIアクションを提供します。クエリの使用はAPI、Verified Access にアクセスする最も直接的な方法です。ただし、この方法では、リクエストに署名するハッシュの生成やエラー処理など、低レベルの詳細な作業をアプリケーションで処理する必要があります。詳細については、「Amazon EC2APIリファレンス」の「Verified Access アクション」を参照してください。

このガイドでは、 を使用して Verified Access リソース AWS Management Console を作成、アクセス、管理する方法について説明します。

料金

Verified Access 上のアプリケーションごとに時間単位で課金され、Verified Access で処理されたデータ量に対して課金されます。詳細については、AWS Verified Access の料金を参照してください。

Verified Access のしくみ

AWS Verified Access は、ユーザーからの各アプリケーションリクエストを評価し、以下に基づいてアクセスを許可します。

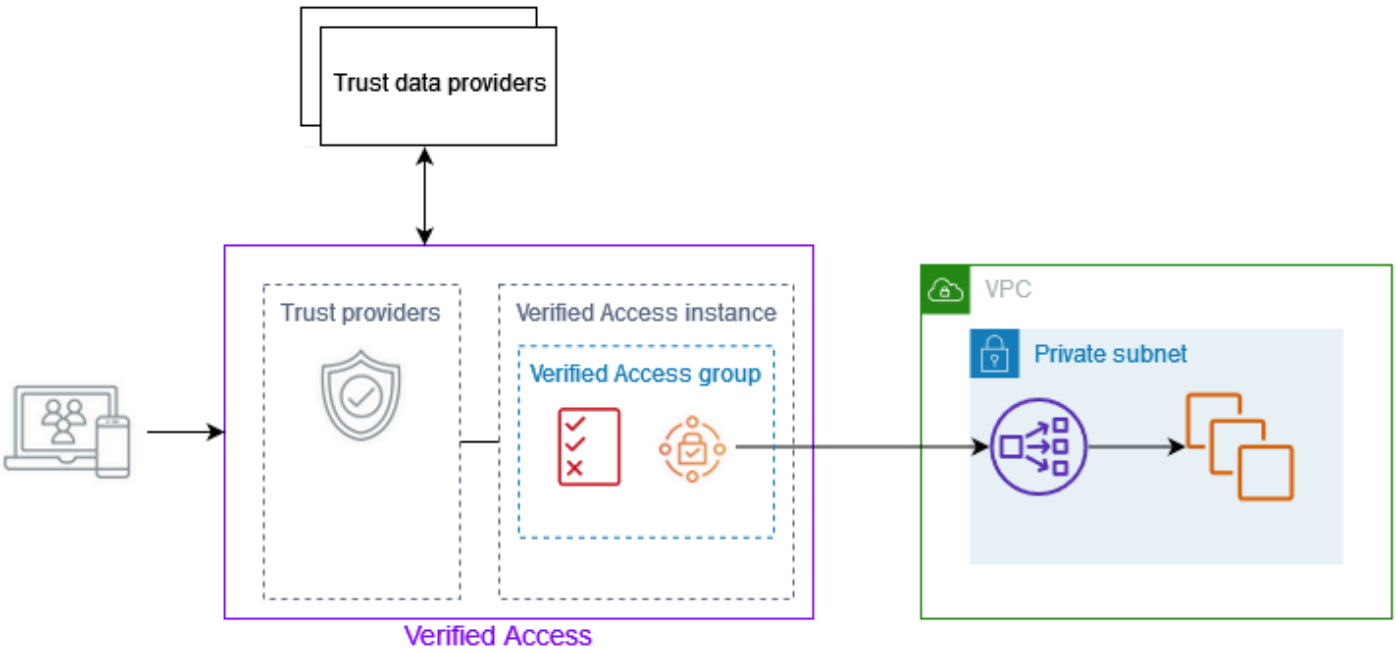
- 選択した信頼プロバイダー (AWS またはサードパーティー) から送信された信頼データ。
- Verified Access で作成したアクセスポリシー。

ユーザーがアプリケーションにアクセスしようとすると、Verified Access は信頼プロバイダーからデータを取得し、そのデータをアプリケーションに設定したポリシーと照らし合わせて評価します。Verified Access は、指定されたセキュリティ要件をユーザーが満たしている場合にのみ、要求されたアプリケーションへのアクセスを許可します。ポリシーが定義されるまで、すべてのアプリケーションリクエストはデフォルトで拒否されます。

さらに、Verified Access はすべてのアクセス試行を口グに記録するため、セキュリティインシデントや監査請求に迅速に対応できます。

Verified Access の主要コンポーネント

次の図は、Verified Access の仕組みの大まかな概要を示しています。ユーザーはアプリケーションへのアクセス要求を送信します。Verified Access は、グループのアクセスポリシーおよびアプリケーション固有のエンドポイントポリシーと照らし合わせてリクエストを評価します。Access が許可されている場合、リクエストはエンドポイントを介してアプリケーションに送信されます。



- Verified Access インスタンス — インスタンスはアプリケーションリクエストを評価し、セキュリティ要件が満たされた場合にのみアクセス権を付与します。
- Verified Access エンドポイント — 各エンドポイントはアプリケーションを表します。ロードバランサーエンドポイントまたはネットワークインターフェースエンドポイントを作成できます。
- Verified Access グループ — Verified Access エンドポイントのコレクション。同様のセキュリティ要件を持つアプリケーションのエンドポイントをグループ化し、ポリシー管理を簡素化することをお勧めします。たとえば、すべての営業アプリケーションのエンドポイントをグループ化できます。
- アクセスポリシー — アプリケーションへのアクセスを許可するか拒否するかを決定するユーザー定義のルール。ユーザー ID やデバイスのセキュリティ状態など、さまざまな要素を組み合わせて指定できます。Verified Access グループごとにグループアクセスポリシーを作成します。このポリシーは、グループ内のすべてのエンドポイントに継承されます。オプションでアプリケーション固有のポリシーを作成し、特定のエンドポイントに添付できます。
- 信頼プロバイダー — ユーザー ID やデバイスのセキュリティ状態を管理するサービス。Verified Access は、 AWS とサードパーティーの信頼プロバイダーの両方で動作します。各 Verified Access インスタンスには少なくとも 1 つの信頼プロバイダーを接続する必要があります。各 Verified Access インスタンスには、1 つの ID 信頼プロバイダーと複数のデバイス信頼プロバイダーを接続できます。
- トラストデータ — 信頼プロバイダーが Verified Access に送信する、ユーザーまたはデバイスのセキュリティ関連データ。ユーザークレームまたは トラストコンテキスト とも呼ばれます。たと

えば、ユーザーの電子メールアドレスやデバイスのオペレーティングシステムバージョンなどです。Verified Access は、アプリケーションへのアクセス要求を受信すると、このデータをアクセスポリシーと照らし合わせて評価します。

チュートリアル: Verified Access の使用を開始する

このチュートリアルを使用して の使用を開始します。 AWS Verified Access。 Verified Access リソースを作成および設定する方法について説明します。

このチュートリアルの一環として、Verified Access にアプリケーションを追加します。チュートリアルを終了すると、特定のユーザーは を使用せずにインターネット経由でそのアプリケーションにアクセスできますVPN。

> **Note**
>
> このチュートリアルでは、デバイスベースの信頼プロバイダーとの統合については説明しません。代わりに、ID ベースの信頼プロバイダーのみを使用します。

タスク

Verified Access チュートリアルの前提条件

このチュートリアルを完了するための前提条件は次のとおりです。

- 2 つの の可用性 AWS アカウント。 1 つのアカウントがターゲットアプリケーションをホストし、Verified Access リソースがもう 1 つのアカウントに作成されます。
- AWS IAM Identity Center で が有効になっている AWS リージョン 作業している 。その後、IAM アイデンティティセンターを Verified Access の信頼プロバイダーとして使用できます。詳細については、「」のIAM「アイデンティティセンターを有効にする」を参照してください。 AWS IAM Identity Center ユーザーガイド。
- パブリックホストドメインと、ドメインのDNSレコードを更新するために必要なアクセス許可。
- の内部ロードバランサーの背後で実行されているアプリケーション AWS アカウント。 使用するアプリケーションドメイン名の例は ですwww.myapp.example.com。
- 自己署名証明書またはパブリックTLS証明書。キーの長さが 1,024 または 2,048 のRSA証明書を使用します。
- の作成に必要なすべてのアクセス許可を持つ IAMポリシー AWS Verified Access ここに書き留めた インスタンスVerified Access インスタンスを作成するためのポリシー。

ステップ 1： Verified Access インスタンスを作成する

以下の手順に従って Verified Access インスタンスを作成します。

Verified Access インスタンスを作成するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. Amazon VPCナビゲーションペインで、Verified Access インスタンス を選択し、Verified Access インスタンス を作成します。
3. (オプション) [名前] と [説明] に、Verified Access インスタンスの名前と説明を入力します。
4. [信頼プロバイダー] については、デフォルトのオプションをそのまま使用してください。
5. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
6. [Verified Access インスタンスの作成] を選択します。

ステップ 2: Verified Access 信頼プロバイダーを設定する

をセットアップできます。 AWS IAM Identity Center を信頼プロバイダーとして。

IAM Identity Center 信頼プロバイダーを作成するには

1. Amazon VPCナビゲーションペインで、Verified Access 信頼プロバイダー を選択し、Verified Access 信頼プロバイダー を作成します。
2. (オプション) [名前タグ] と [説明] に、Verified Access 信頼プロバイダーの名前と説明を入力します。
3. 後でポリシー参照名のポリシールールを利用するときに使用するカスタム ID を入力します。例えば、「**idc**」 と入力します。
4. [信頼プロバイダのタイプ] で、[ユーザー信頼プロバイダー] を選択します。
5. 「ユーザー信頼プロバイダータイプ」で、IAM「アイデンティティセンター」を選択します。
6. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
7. [Verified Access 信頼プロバイダーの作成] を選択します。

ステップ 3: Verified Access インスタンスに信頼プロバイダーをアタッチする

信頼プロバイダーを設定したので、先ほど作成した Verified Access インスタンスにアタッチできます。次の手順に従って、Verified Access インスタンスに信頼プロバイダーを接続します。

信頼プロバイダーをインスタンスに接続するには

1. Amazon VPCナビゲーションペインで、Verified Access インスタンス を選択します。
2. インスタンスを選択します。
3. [アクション]、[Verified Access 信頼プロバイダーを添付] を選択します。
4. [Verified Access 信頼プロバイダー] で、信頼プロバイダーを選択します。
5. [Verified Access 信頼プロバイダーを添付] を選択します。

ステップ 4: Verified Access グループを作成する

このステップでは、ステップ 5 でエンドポイントとして使用するグループを作成します。

Verified Access グループを作成するには

1. Amazon VPCナビゲーションペインで、検証済みアクセスグループ を選択し、検証済みアクセスグループ を作成します。
2. (オプション) [名前タグ] と [説明] に、グループの名前と説明を入力します。
3. [Verified Access インスタンス] で、Verified Access インスタンスを選択します。
4. [ポリシー定義] については、空白のままにしてください。このチュートリアルの後半で、ポリシーを作成します。
5. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
6. [Verified Access グループの作成] を選択します。

ステップ 5: を使用して Verified Access グループを共有する AWS Resource Access Manager

このステップでは、先ほど作成したグループを と共有します。 AWS アカウント ターゲットアプリケーションが実行されている 。Verified Access グループを共有するには、リソース共有に追加する必要があります。リソース共有がない場合は、まずリソース共有を作成する必要があります。

の組織に属している場合 AWS Organizations、および組織内での共有が有効になっている場合、組織内のコンシューマーには、共有 Verified Access グループへのアクセス権が自動的に付与されます。これに該当しない場合、コンシューマーはリソース共有への参加の招待を受け取り、その招待を受け入れた後で、共有 Verified Access グループに対するアクセス許可が付与されます。

の「リソース共有の作成」のステップに従います。 AWS RAM ユーザーガイド。[リソースタイプを選択]で、[Verified Access グループ] を選択し、Verified Access グループのチェックボックスを選択します。

詳細については、「」の「開始方法」を参照してください。 AWS RAM ユーザーガイド。

ステップ 6: Verified Access エンドポイントを作成してアプリケーションを追加する

Verified Access エンドポイントを作成するには、次の手順に従います。このステップは、Elastic Load Balancing の内部ロードバランサーの背後でアプリケーションを実行していることを前提としています。

Verified Access エンドポイントを作成するには

1. Amazon VPCナビゲーションペインで、Verified Access エンドポイント を選択し、次に Verified Access エンドポイント を作成します。
2. (オプション) [名前タグ] と [説明] に、エンドポイントの名前と説明を入力します。
3. [Verified Access グループ] では、Verified Access グループを選択します。
4. [アプリケーション詳細] では、次の操作を行います。
   a. アプリケーションドメイン に、アプリケーションDNSの名前を入力します。
   b. ドメイン証明書 ARNで、パブリックTLS証明書の Amazon リソースネーム (ARN) を選択します。
5. [エンドポイント詳細] では、次の操作を行います。
   a. 添付ファイルタイプ で、 を選択しますVPC。
   b. (オプション) [セキュリティグループ]で、エンドポイントに関連付けるセキュリティグループを選択します。
   c. [エンドポイントドメインプレフィックス] には、カスタム ID を入力します。これは、Verified Access が生成するDNS名前の前に付加されます。この例では、**`my-ava-app`** を使用します。
   d. [エンドポイントタイプ] で、[ロードバランサー] を選択します。
   e. プロトコル で、 HTTPSまたは を選択しますHTTP。これはロードバランサーの設定によって異なります。
   f. [ポート] に、ポート番号を入力します。これはロードバランサーの設定によって異なります。
   g. ロードバランサー ARNで、ロードバランサーを選択します。
   h. [サブネット] では、ロードバランサーに関連付けられているサブネットを選択します。
6. [ポリシー定義] には、現時点ではポリシーを入力しないでください。これについては、このチュートリアルの後半で説明します。
7. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
8. [Verified Access エンドポイントの作成] を選択します。

ステップ 7: Verified Access エンドポイントDNSの設定を構成する

このステップでは、アプリケーションのドメイン名 (www.myapp.example.com など) を Verified Access エンドポイントのドメイン名にマッピングします。DNS マッピングを完了するには、DNSプロバイダーで正規名レコード (CNAME) を作成します。CNAME レコードを作成すると、ユーザーからアプリケーションへのすべてのリクエストが Verified Access に送信されます。

エンドポイントの ドメイン名を入手するためには

1. Amazon VPCナビゲーションペインで、Verified Access エンドポイント を選択します。
2. 前に作成したエンドポイントを選択します。
3. エンドポイントの [詳細] タブを選択します。
4. エンドポイントドメイン で、エンドポイントドメインをコピーします。

このチュートリアルでは、エンドポイントのドメイン名は「`my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`」になります。

DNS プロバイダーでCNAMEレコードを作成します。

レコード名	タイプ	値
www.myapp.example.com	CNAME	my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com

ステップ 8: Verified Access に追加したアプリケーションへの接続をテストする

これで、アプリケーションへの接続をテストできます。アプリケーションのドメイン名をウェブブラウザに入力します。Verified Access ポリシーのデフォルト動作では、すべてのリクエストが拒否されます。誰もがアクセスできるようにするポリシーをまだ制定していないため、すべてのリクエストは拒否されるはずです。

ステップ 9: Verified Access グループレベルのアクセスポリシーを設定する

以下の手順に従って Verified Access グループを変更し、アプリケーションへの接続を許可するアクセスポリシーを設定します。ポリシーの詳細は、IAMIdentity Center で設定されているユーザーとグループによって異なります。ポリシーの作成の詳細については、「Verified Access ポリシー」を参照してください。

Verified Access グループを変更するには

1. Amazon VPCナビゲーションペインで、Verified Access グループ を選択します。
2. グループを選択します。
3. [アクション]、[Verified Access グループポリシーの変更] を選択します。
4. ポリシーを入力します。
5. [Verified Access グループポリシーの変更] を選択します。

ステップ 10: Verified Access に追加したアプリケーションへの接続を再テストする

グループポリシーが設定されたので、アプリケーションにアクセスできます。アプリケーションのドメイン名をウェブブラウザに入力します。リクエストが許可され、アプリケーションにリダイレクトされるはずです。

作成した Verified Access リソースをクリーンアップする

テストが終了したら、以下の手順に従って作成されたリソースを削除します。

このチュートリアルで作成した Verified Access リソースを削除するには

1. Amazon VPCナビゲーションペインで、Verified Access エンドポイント を選択します。削除するエンドポイントを選択します。[アクション]、[Verified Access エンドポイントの削除] を選択します。
2. ナビゲーションペインで、[Verified Access グループ] を選択します。削除するグループを選択します。[アクション]、[Verified Access グループの削除] を選択します。注 - エンドポイントの削除処理が完了するまで数分かかる場合があります。

3. Amazon VPCナビゲーションペインで、Verified Access インスタンス を選択します。このチュートリアル用に作成したインスタンスを選択します。[アクション]、[Verified Access 信頼プロバイダーを切り離す] を選択します。ドロップダウンリストから信頼プロバイダーを選択し、[Verified Access 信頼プロバイダーを切り離す] を選択します。
4. Amazon VPCナビゲーションペインで、Verified Access 信頼プロバイダー を選択します。このチュートリアルで作成した信頼プロバイダーを選択します。[アクション]、[Verified Access 信頼プロバイダーの削除] を選択します。
5. Amazon VPCナビゲーションペインで、Verified Access インスタンス を選択します。このチュートリアル用に作成したインスタンスを選択します。[アクション]、[Verified Acess インスタンスの削除] を選択します。

Verified Access インスタンス

AWS Verified Access インスタンスは、信頼プロバイダーと Verified Access グループの整理に役立つ AWS リソースです。インスタンスはアプリケーションリクエストを評価し、セキュリティ要件が満たされた場合にのみアクセスを許可します。

トピック

Verified Access インスタンスの作成と管理

Verified Access インスタンスを使用して、信頼プロバイダーと Verified Access グループを整理します。以下の手順を使用して Verified Access インスタンスを作成し、信頼プロバイダーを Verified Access にアタッチするか、Verified Access から信頼プロバイダーをデタッチします。

トピック

Verified Access インスタンスの作成

以下の手順に従って Verified Access インスタンスを作成します。

Verified Access インスタンスを作成するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで [Verified Access インスタンス] を選択し、[Verified Access インスタンスの作成] を選択します。
3. (オプション) [名前] と [説明] に、Verified Access インスタンスの名前と説明を入力します。
4. （ オプション) Verified Access に準拠する必要がある場合は、連邦情報プロセス標準 (FIPS) FIPS の有効化を選択します。

5. (オプション) [信頼プロバイダー] では、Verified Access インスタンスに添付する信頼プロバイダーを選択します。
6. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
7. [Verified Access インスタンスの作成] を選択します。

Verified Access インスタンスに信頼プロバイダーをアタッチする

以下の手順に従って Verified Access インスタンスに信頼プロバイダーを添付します。

Verified Access インスタンスに信頼プロバイダーを添付するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. インスタンスを選択します。
4. [アクション]、[Verified Access 信頼プロバイダーを添付] を選択します。
5. [Verified Access 信頼プロバイダー] では、信頼プロバイダーを選択します。
6. [Verified Access 信頼プロバイダーを添付] を選択します。

Verified Access インスタンスから信頼プロバイダーをデタッチする

以下の手順に従って、Verified Access インスタンスから信頼プロバイダーを切り離すします。

Verified Access インスタンスから信頼プロバイダーを切り離すには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. インスタンスを選択します。
4. [アクション]、[Verified Access 信頼プロバイダーを切り離す] を選択します。
5. [Verified Access 信頼プロバイダー] で、信頼プロバイダーを選択します。
6. [Verified Access 信頼プロバイダーを切り離す] を選択します。

Verified Access インスタンスの削除

浮揚になった Verified Access インスタンスは、削除することができます。インスタンスを削除する前に、関連付けられている信頼プロバイダーまたは Verified Access グループをすべて削除する必要があります。

Verified Access インスタンスを削除するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [アクション]、[Verified Acess インスタンスの削除] を選択します。
5. 確認を求められたら、「**delete**」と入力してから、[削除] を選択します。

Verified Access を と統合する AWS WAF

Verified Access によって適用される認証ルールと承認ルールに加えて、ペリメータ保護を適用する場合もあります。これにより、アプリケーションを他の脅威から保護することができます。これは、Verified Access デプロイ AWS WAF に統合することで実現できます。 AWS WAF は、保護されたウェブアプリケーションリソースに転送される HTTP(S) リクエストをモニタリングできるウェブアプリケーションファイアウォールです。の詳細については AWS WAF、「 AWS WAF デベロッパーガイドAWS WAF」の「」を参照してください。

AWS WAF ウェブアクセスコントロールリスト (ACL) を Verified Access インスタンスに関連付けることで、Verified Access AWS WAF と統合できます。ウェブACLは、保護された AWS WAF リソースが応答するすべての HTTP(S) ウェブリクエストをきめ細かく制御できるリソースです。 AWS WAF 関連付けまたは関連付け解除リクエストの処理中に、インスタンスにアタッチされた Verified Access エンドポイントのステータスが として表示されますupdating。リクエストが完了すると、ステータスは `active` に戻ります。ステータスは、 AWS Management Console または でエンドポイントを記述することで表示できます AWS CLI。

> **Note**
>
> AWS WAF コンソールまたは を使用してAPI、この統合を実行することもできます。Verified Access インスタンスの Amazon リソースネーム (ARN) が必要になります。これは、

次の形式ARNを使用して構築できます: `arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}`。

トピック

- IAM Verified Access を と統合するために必要な アクセス許可 AWS WAF
- AWS WAF ウェブを関連付ける ACL
- AWS WAF 統合のステータスを確認する
- AWS WAF ウェブの関連付けを解除する ACL

IAM Verified Access を と統合するために必要な アクセス許可 AWS WAF

Verified Access AWS WAF との統合には、 APIオペレーションに直接対応しないアクセス許可のみのアクションが含まれます。これらのアクションについては、「[permission only] による AWS Identity and Access Management サービス認証リファレンス」に記載されています。「サービス認証リファレンス」の「Amazon のアクション、リソース、および条件キーEC2」を参照してください。

ウェブ を使用するにはACL、 AWS Identity and Access Management プリンシパルに次のアクセス許可が必要です。

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

AWS WAF ウェブを関連付ける ACL

次の手順は、 を使用して AWS WAF ウェブアクセスコントロールリスト (ACL) を Verified Access インスタンスに関連付ける方法を示しています AWS Management Console。

> **Tip**
>
> 以下の手順ACLを完了するには、既存の AWS WAF ウェブが必要です。ウェブの詳細については、「 AWS WAF デベロッパーガイド」の「ウェブアクセスコントロールリストACLs」を参照してください。

AWS WAF ウェブを ACL Verified Access インスタンスに関連付けるには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [統合] タブを選択します。
5. アクション を選択し、ウェブ を関連付けますACL。
6. ウェブ ACLで、既存のウェブ を選択しACL、ウェブ の関連付けを選択しますACL。

AWS Management Console の を使用して AWS WAF 、このタスクを実行することもできます。詳細については、「 デベロッパーガイド」の「 AWSリソースACLとのウェブの関連付けまたは関連付け解除」を参照してください。 AWS WAF

AWS WAF 統合のステータスを確認する

を使用して、 AWS WAF ウェブアクセスコントロールリスト (ACL) が Verified Access インスタンスに関連付けられているかどうかを確認できます AWS Management Console。

Verified Access インスタンスと AWS WAF の統合のステータスを表示するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [統合] タブを選択します。
5. WAF 統合ステータス にリストされている詳細を確認します。ステータスは、関連状態の場合、ウェブACL識別子とともに関連 または関連なし として表示されます。

AWS WAF ウェブの関連付けを解除する ACL

次の手順は、 を使用して AWS WAF ウェブアクセスコントロールリスト (ACL) と Verified Access インスタンスの関連付けを解除する方法を示しています AWS Management Console。

Verified Access インスタンスACLから AWS WAF ウェブの関連付けを解除するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [統合] タブを選択します。
5. アクション を選択し、ウェブ の関連付けを解除しますACL。
6. ウェブ の関連付けを解除ACLを選択して確認します。

AWS Management Console の を使用して AWS WAF 、このタスクを実行することもできます。詳細については、「 デベロッパーガイド」の「 AWSリソースACLとのウェブの関連付けまたは関連付け解除」を参照してください。 AWS WAF

FIPS Verified Access の コンプライアンス

連邦情報処理規格 (FIPS) は、機密情報を保護する暗号化モジュールのセキュリティ要件を指定する米国およびカナダ政府の規格です。 は、 FIPS刊行物 140-2 に準拠するように環境を設定するオプション AWS Verified Access を提供します。FIPS Verified Access の コンプライアンスは、次の AWS リージョンで利用できます。

- 米国東部(オハイオ)
- 米国東部 (バージニア北部)
- 米国西部 (北カリフォルニア)
- 米国西部 (オレゴン)
- カナダ (中部)
- AWS GovCloud (US) 西部
- AWS GovCloud (US) 東部

このページでは、新規または既存の Verified Access 環境をFIPS準拠するように設定する方法を示します。

トピック

FIPS コンプライアンスのために既存の Verified Access 環境を設定する

既存の Verified Access 環境があり、FIPS準拠するように設定する場合は、FIPSコンプライアンスを有効にするために一部のリソースを削除して再作成する必要があります。

既存の AWS Verified Access 環境をFIPS準拠するように再設定するには、以下の手順に従います。

1. 元の Verified Access エンドポイント、グループ、インスタンスを削除します。設定した信頼プロバイダーは再利用できます。
2. Verified Access インスタンスを作成し、作成時に連邦情報プロセス標準 (FIPS) を必ず有効にします。また、作成時に、使用する [Verified Access 信頼プロバイダー] をドロップダウンリストから選択して添付します。
3. Verified Access グループを作成します。グループの作成時、作成したばかりの Verified Access インスタンスにそのグループを関連付けます。
4. 1 つ以上の Verified Access エンドポイント を作成します。エンドポイントの作成時に、前のステップで作成したグループにエンドポイントを関連付けます。

FIPS コンプライアンスのために新しい Verified Access 環境を設定する

FIPS 準拠する新しい AWS Verified Access 環境を設定するには、以下のステップに従います。

1. 信頼プロバイダーを設定します。必要に応じて、ユーザー ID 信頼プロバイダー、(オプションで) デバイスベースの 信頼プロバイダーを作成する必要があります。
2. Verified Access インスタンス を作成し、プロセス中に連邦情報プロセス標準 (FIPS) を必ず有効にします。また、作成時に、前のステップで作成した Verified Access 信頼プロバイダー をドロップダウンリストから選択して添付します。
3. Verified Access グループを作成します。グループの作成時、作成したばかりの Verified Access インスタンスにそのグループを関連付けます。
4. 1 つ以上の Verified Access エンドポイント を作成します。エンドポイントの作成時に、前のステップで作成したグループにエンドポイントを関連付けます。

Verified Access 信頼プロバイダー

信頼プロバイダーは、ユーザーとデバイスに関する情報を に送信するサービスです AWS Verified Access。この情報はトラストコンテキストと呼ばれます。これには、メールアドレスや「営業」組織のメンバーなどのユーザー ID に基づく属性や、インストール済みのセキュリティパッチやウイルス対策ソフトウェアのバージョンなどのデバイス管理情報が含まれる場合があります。

Verified Access は、以下のカテゴリの信頼プロバイダーをサポートします。

- ユーザー ID — ユーザーのデジタルアイデンティティを保存および管理する ID プロバイダー (IdP) サービス。
- デバイス管理 — ラップトップ、タブレット、スマートフォンなどのデバイス用のデバイス管理システム。

内容

Verified Access のユーザー ID 信頼プロバイダー

AWS IAM Identity Center または OpenID Connect 互換のユーザー ID 信頼プロバイダーのいずれかを使用できます。

内容

IAM Identity Center を信頼プロバイダーとして使用する

AWS Verified Access では、ユーザー ID 信頼プロバイダー AWS IAM Identity Center として を使用できます。

前提条件と考慮事項

- IAM Identity Center インスタンスは AWS Organizations インスタンスである必要があります。スタンドアロン AWS アカウントの IAM Identity Center インスタンスは機能しません。

- IAM Identity Center インスタンスは、Verified Access 信頼プロバイダーを作成するリージョンと同じ AWS リージョンで有効にする必要があります。

さまざまなインスタンスタイプの詳細については、「 ユーザーガイド」のIAM「アイデンティティセンターの組織インスタンスとアカウントインスタンスの管理」を参照してください。 AWS IAM Identity Center

IAM Identity Center 信頼プロバイダーを作成する

AWS アカウントで IAM Identity Center を有効にしたら、次の手順を使用して、Verified Access の信頼プロバイダーとして IAM Identity Center を設定できます。

IAM Identity Center 信頼プロバイダーを作成するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、「 Verified Access 信頼プロバイダー」を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、信頼プロバイダーの名前と説明を入力します。
4. [ポリシー参照名] には、後でポリシールールを利用するときに使用する識別子を入力します。
5. [信頼プロバイダのタイプ] で、[ユーザー信頼プロバイダー] を選択します。
6. 「ユーザー信頼プロバイダータイプ」で、IAM「アイデンティティセンター」を選択します。
7. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
8. [Verified Access 信頼プロバイダーの作成] を選択します。

IAM Identity Center 信頼プロバイダーを作成するには (AWS CLI)

- create-verified-access-trustプロバイダー (AWS CLI)

IAM Identity Center 信頼プロバイダーを削除する

信頼プロバイダーを削除する前に、信頼プロバイダーが添付されているインスタンスからすべてのエンドポイントとグループ設定を削除する必要があります。

IAM Identity Center 信頼プロバイダーを削除するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。

2. ナビゲーションペインで [Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダー] で削除する信頼プロバイダーを選択します。
3. 「アクション」、「 Verified Access 信頼プロバイダの削除」の順に選択します。
4. テキストボックスに「delete」と入力して削除を確定します。
5. [削除] を選択します。

IAM Identity Center 信頼プロバイダーを削除するには (AWS CLI)

- delete-verified-access-trustプロバイダー (AWS CLI)

OpenID Connect 信頼プロバイダーを使用する

AWS Verified Access は、標準の OpenID Connect (OIDC) メソッドを使用する ID プロバイダーをサポートします。Verified Access では、OIDC互換性のあるプロバイダーをユーザー ID 信頼プロバイダーとして使用できます。ただし、潜在的なOIDCプロバイダーが多数存在するため、 AWS は Verified Access との各OIDC統合をテストできません。

Verified Access は、OIDCプロバイダーの から評価される信頼データを取得しますUserInfo Endpoint。この Scope パラメータは、検索するトラストデータのセットを決定するために使用されます。トラストデータを受信すると、Verified Access ポリシーがそのデータに対して評価されます。

> **Note**
>
> Verified Access は、Verified Access ポリシーを評価する際に、OIDCプロバイダーからID token送信された からの信頼データを使用しません。UserInfo Endpoint からのトラストデータのみがポリシーに照らして評価されます。

内容

- OIDC 信頼プロバイダーを作成するための前提条件
- OIDC 信頼プロバイダーを作成する
- OIDC 信頼プロバイダーを変更する
- OIDC 信頼プロバイダーを削除する

OIDC 信頼プロバイダーを作成するための前提条件

信頼プロバイダーサービスから次の情報を直接収集する必要があります。

- Issuer
- 認可エンドポイント
- トークンエンドポイント
- UserInfo エンドポイント
- クライアント ID
- クライアントシークレット
- スコープ

OIDC 信頼プロバイダーを作成する

信頼プロバイダーOIDCとして を作成するには、次の手順に従います。

OIDC 信頼プロバイダーを作成するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、「 Verified Access 信頼プロバイダー」を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、信頼プロバイダーの名前と説明を入力します。
4. [ポリシー参照名] には、後でポリシールールを利用するときに使用する識別子を入力します。
5. [信頼プロバイダのタイプ] で、[ユーザー信頼プロバイダー] を選択します。
6. ユーザー信頼プロバイダータイプ で、 OIDC (OpenID Connect) を選択します。
7. 発行者 には、OIDC発行者の識別子を入力します。
8. 認証エンドポイント には、認証エンドポイントURLの完全な を入力します。
9. トークンエンドポイント には、トークンエンドポイントURLの完全な を入力します。
10. ユーザーエンドポイント には、ユーザーエンドポイントURLの完全な を入力します。
11. クライアント ID OAuth の 2.0 クライアント識別子を入力します。
12. クライアントシークレット OAuth の 2.0 クライアントシークレットを入力します。
13. ID プロバイダーで定義されている対象範囲のスペースで区切られたリストを入力します。対象範囲には少なくとも「openid」対象範囲が必要です。

14. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
15. [Verified Access 信頼プロバイダーの作成] を選択します。

> **Note**
>
> OIDC プロバイダーの許可リストURIにリダイレクトを追加する必要があります。このためは、Verified Access エンドポイントの `ApplicationDomain` を使用します。これは AWS Management Console、、Verified Access エンドポイントの詳細タブ、または を使用してエンドポイントを AWS CLI 記述することで確認できます。OIDC プロバイダーの許可リストに `https://ApplicationDomain/oauth2/idpresponse` を追加します。

OIDC 信頼プロバイダーを作成するには (AWS CLI)

- create-verified-access-trustプロバイダー (AWS CLI)

OIDC 信頼プロバイダーを変更する

信頼プロバイダーの作成後、その設定を更新できます。

OIDC 信頼プロバイダーを変更するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで [Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダー] で変更する信頼プロバイダーを選択します。
3. [アクション]、[Verified Access 信頼プロバイダの変更] の順に選択します。
4. オプションを変更します。
5. [Verified Access 信頼プロバイダーの変更] を選択します。

OIDC 信頼プロバイダーを変更するには (AWS CLI)

- modify-verified-access-trustプロバイダー (AWS CLI)

OIDC 信頼プロバイダーを削除する

ユーザーの信頼プロバイダーを削除する前に、まず信頼プロバイダーが添付されているインスタンスからすべてのエンドポイントとグループ設定を削除する必要があります。

OIDC 信頼プロバイダーを削除するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで [Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダー] で削除する信頼プロバイダーを選択します。
3. 「アクション」、「 Verified Access 信頼プロバイダの削除」の順に選択します。
4. テキストボックスに「`delete`」と入力して削除を確定します。
5. [削除] を選択します。

OIDC 信頼プロバイダーを削除するには (AWS CLI)

- delete-verified-access-trustプロバイダー (AWS CLI)

Verified Access 用のデバイスベースの信頼プロバイダー

AWS Verified Access でデバイス信頼プロバイダーを使用できます。Verified Access インスタンスでは 1 つまたは複数のデバイス信頼プロバイダーを使用できます。

内容

サポートされているデバイス信頼プロバイダー

以下のデバイス信頼プロバイダーは Verified Access と統合できます。

- CrowdStrike – CrowdStrike および Verified Access によるプライベートアプリケーションの保護
- Jamf — Verified Access と Jamf デバイス ID の統合
- JumpCloud – 統合 JumpCloud と AWS 検証済みアクセス

デバイスベースの信頼プロバイダーの作成

これらの手順に従って、Verified Access で使用するデバイス信頼プロバイダーを作成して設定します。

Verified Access デバイスの信頼プロバイダーを作成するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、「 Verified Access 信頼プロバイダー」を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、信頼プロバイダーの名前と説明を入力します。
4. ポリシー参照名のポリシールールを後で利用する際に使用する識別子を入力します。
5. [信頼プロバイダーのタイプ] には、[デバイス ID] を選択します。
6. デバイス ID タイプ で、Jamf 、CrowdStrike、または を選択しますJumpCloud。
7. [テナント ID] には、テナントアプリケーションの識別子を入力します。
8. (オプション) パブリック署名キー URLに、デバイスの信頼プロバイダーによってURL共有されている一意のキーを入力します。(このパラメータは Jamf CrowdStrike または Jumpcloud には必要ありません)。
9. [Verified Access 信頼プロバイダーの作成] を選択します。

> **Note**
>
> OIDC プロバイダーの許可リストURIにリダイレクトを追加する必要があります。このためは、Verified Access エンドポイントの `DeviceValidationDomain` を使用します。これは AWS Management Console、Verified Access エンドポイントの詳細タブの 、または を使用してエンドポイントを AWS CLI 記述することで確認できます。OIDC プロバイダーの許可リストに https://`DeviceValidationDomain`/oauth2/idpresponse を追加します。

Verified Access デバイス信頼プロバイダーを作成するには (AWS CLI)

- create-verified-access-trustプロバイダー (AWS CLI)

デバイスベースの信頼プロバイダーの変更

信頼プロバイダーの作成後、その設定を更新できます。

Verified Access デバイスの信頼プロバイダーを変更するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access 信頼プロバイダー] を選択します。
3. 信頼プロバイダーを選択します。
4. [アクション] を選択し、[Verified Access 信頼プロバイダーの変更] を選択します。
5. 必要に応じて説明を変更します。
6. (オプション) パブリック署名キー URLで、デバイス信頼プロバイダーによってURL共有される一意のキーを変更します。(このパラメータは、デバイス信頼プロバイダーが Jamf、CrowdStrike または Jumpcloud の場合は必要ありません)。
7. [Verified Access 信頼プロバイダーの変更] を選択します。

Verified Access デバイスの信頼プロバイダーを変更するには (AWS CLI)

- modify-verified-access-trustプロバイダー (AWS CLI)

デバイスベースの信頼プロバイダーの削除

不要になった信頼プロバイダーは、削除することができます。

Verified Access デバイスの信頼プロバイダーを削除するには (AWS コンソール)

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access 信頼プロバイダー] を選択します。
3. 「Verified Access 信頼プロバイダー」で、削除する信頼プロバイダーを選択します。
4. [アクション] を選択し、[Verified Access 信頼プロバイダーの削除] を選択します。
5. 確認を求められたら、「**`delete`**」と入力してから、[Delete] (削除) を選択します。

Verified Access デバイス信頼プロバイダーを削除するには (AWS CLI)

- delete-verified-access-trustプロバイダー (AWS CLI)

Verified Access グループ

An AWS Verified Access group は、Verified Access エンドポイントとグループレベルの Verified Access ポリシーのコレクションです。グループ内の各エンドポイントは、Verified Access ポリシーを共有します。グループを使用して、共通のセキュリティ要件を持つエンドポイントをまとめることができます。これにより、1 つのポリシーで複数のアプリケーションのセキュリティニーズに対応できるため、ポリシー管理の簡素化に役立ちます。

たとえば、すべての営業アプリケーションをグループ化して、グループ全体のアクセスポリシーを設定できます。その後、このポリシーを使用して、すべての営業アプリケーションに共通の最低限のセキュリティ要件を定義できます。このアプローチは、ポリシー管理の簡素化に役立ちます。

グループを作成する際、グループを Verified Access インスタンスに関連付ける必要があります。エンドポイントを作成する過程で、エンドポイントをグループに関連付けます。

タスク

- Verified Access グループの作成
- Verified Access グループポリシーの変更
- Verified Access グループを削除する

Verified Access グループの作成

以下の手順に従って、Verified Access グループを作成します。

Verified Access グループを作成するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで, [Verified Access グループ] を選択し、次に [Verified Access グループの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、グループの名前と説明を入力します。
4. [Verified Access インスタンス] には、グループに関連付ける Verified Access インスタンスを選択します。
5. (オプション)[ポリシー定義] には、グループに適用する Verified Access ポリシーを入力します。
6. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。

7. [Verified Access グループの作成] を選択します。

Verified Access グループポリシーの変更

Verified Access グループのポリシーを変更するには、次の手順に従います。変更が有効になるまでに数分かかります。

Verified Access グループポリシーを変更するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access グループ] を選択します。
3. グループを選択します。
4. [アクション]、[Verified Access グループポリシーの変更] を選択します。
5. （ オプション) 必要に応じてポリシーの有効化をオンまたはオフにします。
6. （ オプション) ポリシー に、グループに適用する Verified Access ポリシーを入力します。
7. [Verified Access グループポリシーの変更] を選択します。

Verified Access グループを削除する

不要になった Verified Access グループは、削除することができます。

Verified Access グループを削除するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access グループ] を選択します。
3. グループを選択します。
4. [アクション]、[Verified Access グループの削除] を選択します。
5. 確認を求められたら、「**delete**」と入力してから、[削除] を選択します。

Verified Access エンドポイント

Verified Access エンドポイントはアプリケーションを表します。各エンドポイントは Verified Access グループに関連付けられ、グループのアクセスポリシーを継承します。オプションで、アプリケーション固有のエンドポイントポリシーを各エンドポイントに添付することができます。

内容

Verified Access エンドポイントタイプ

Verified Access エンドポイントのタイプは次のとおりです。

- ロードバランサー — アプリケーションリクエストはロードバランサーに送信され、アプリケーションに配布されます。
- ネットワークインターフェース — アプリケーションリクエストは、指定されたプロトコルとポートを使用してネットワークインターフェースに送信されます。

Verified Access が共有サブネットVPCsとサブネットと連携する方法

共有VPCサブネットに関する動作は次のとおりです。

- Verified Access エンドポイントはVPC、サブネット共有でサポートされています。参加者は共有サブネットに Verified Access エンドポイントを作成できます。
- エンドポイントを作成した参加者がエンドポイントの所有者となり、エンドポイントを変更できるのはその参加者だけです。VPC 所有者はエンドポイントを変更できません。

- Verified Access エンドポイントを で作成できない AWS ローカルゾーン、つまりローカルゾーンを介した共有はできません。

詳細については、「Amazon ユーザーガイド」の「 を他の アカウントVPCと共有する」を参照してください。 VPC

Verified Access 用のロードバランサーエンドポイントを作成する

Verified Access のロードバランサーエンドポイントを作成するには、次の手順に従います。ロードバランサーの詳細については、「Elastic Load Balancing ユーザーガイド」を参照してください。

要件

- IPv4 トラフィックのみがサポートされます。
- HTTP および HTTPSプロトコルのみがサポートされています。HTTPS 接続などの存続期間の長い WebSocket 接続はサポートされていません。
- ロードバランサーは、Application Load Balancer または Network Load Balancer のいずれかで、内部ロードバランサーである必要があります。
- ロードバランサーとサブネットは、同じ仮想プライベートクラウド () に属している必要がありますVPC。
- HTTPS ロードバランサーは、自己署名証明書またはパブリックTLS証明書を使用できます。キーの長さが 1,024 または 2,048 のRSA証明書を使用します。
- アプリケーションのドメイン名を入力する必要があります。これは、ユーザーがアプリケーションにアクセスするために使用するパブリックDNS名です。また、このドメイン名に一致する CN を含むパブリックSSL証明書を提供する必要があります。を使用して証明書を作成またはインポートできます。 AWS Certificate Manager.

ロードバランサーエンドポイントを作成するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. [Verified Access エンドポイントの作成] を選択します。
4. (オプション) [名前タグ] と [説明] に、エンドポイントの名前と説明を入力します。
5. [Verified Access グループ] では、エンドポイントの Verified Access グループを選択します。

6. [アプリケーション詳細] では、次の操作を行います。
   a. アプリケーションドメイン に、アプリケーションDNSの名前を入力します。
   b. ドメイン証明書 ARNで、パブリックTLS証明書を選択します。
7. [エンドポイント詳細] では、次の操作を行います。
   a. 添付ファイルタイプ で、 を選択しますVPC。
   b. [セキュリティグループ] で、VPC エンドポイントのセキュリティグループを選択します。Verified Access エンドポイントからロードバランサーに入るトラフィックは、このセキュリティグループに関連付けられます。
   c. エンドポイントドメインプレフィックス には、Verified Access がエンドポイントに対して生成するDNS名前の前に追加するカスタム識別子を入力します。
   d. [エンドポイントタイプ] で、[ロードバランサー] を選択します。
   e. プロトコル で、 HTTPSまたは を選択しますHTTP。
   f. [ポート] に、ポート番号を入力します。
   g. ロードバランサー ARNで、ロードバランサーを選択します。
   h. [サブネット] で、ロードバランサーのサブネットを選択します。
8. (オプション) [ポリシー定義] には、エンドポイントの Verified Access ポリシーを入力します。
9. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
10. [Verified Access エンドポイントの作成] を選択します。

Verified Access のネットワークインターフェイスのエンドポイントを作成する

次の手順に従って、 ネットワークインターフェイスエンドポイントを作成します。

要件

- IPv4 トラフィックのみがサポートされます。
- HTTP および HTTPSプロトコルのみがサポートされています。
- ネットワークインターフェイスは、セキュリティグループと同じ仮想プライベートクラウド (VPC) に属している必要があります。
- ネットワークインターフェースのプライベート IP を使用してトラフィックを転送します。

- アプリケーションのドメイン名を入力する必要があります。これは、ユーザーがアプリケーションにアクセスするために使用するパブリックDNS名です。また、このドメイン名に一致する CN を含むパブリックSSL証明書を提供する必要があります。を使用して証明書を作成またはインポートできます。 AWS Certificate Manager.

ネットワークインターフェイスエンドポイントを作成するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. [Verified Access エンドポイントの作成] を選択します。
4. (オプション) [名前タグ] と [説明] に、エンドポイントの名前と説明を入力します。
5. [Verified Access グループ] では、エンドポイントの Verified Access グループを選択します。
6. [アプリケーション詳細] では、次の操作を行います。
   a. アプリケーションドメイン に、アプリケーションDNSの名前を入力します。
   b. ドメイン証明書 ARNで、パブリックTLS証明書を選択します。
7. [エンドポイント詳細] では、次の操作を行います。
   a. 添付ファイルタイプ で、 を選択しますVPC。
   b. [セキュリティグループ] で、VPC エンドポイントのセキュリティグループを選択します。Verified Access エンドポイントからネットワークインターフェースに入るトラフィックは、このセキュリティグループに関連付けられます。
   c. エンドポイントドメインプレフィックス には、Verified Access がエンドポイントに対して生成するDNS名前の前に追加するカスタム識別子を入力します。
   d. [エンドポイントタイプ] で、[ネットワークインターフェイス] を選択します。
   e. プロトコル で、 HTTPSまたは を選択しますHTTP。
   f. [ポート] に、ポート番号を入力します。
   g. [ネットワークインターフェイス] でネットワークインターフェイスを選択します。
8. (オプション) [ポリシー定義] には、エンドポイントの Verified Access ポリシーを入力します。
9. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
10. [Verified Access エンドポイントの作成] を選択します。

Verified Access エンドポイントから発信されるトラフィックを許可する

Verified Access エンドポイントから発信されるトラフィックを許可するように、アプリケーションのセキュリティグループを設定できます。そのためには、エンドポイントのセキュリティグループをソースとして指定するインバウンドルールを追加します。アプリケーションが Verified Access エンドポイントからのトラフィックのみを受信するように、その他のインバウンドルールを削除することをお勧めします。

既存のアウトバウンドルールは維持することをお勧めします。

アプリケーションのセキュリティグループルールを更新するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. Verified Access エンドポイントを選択し、詳細タブでセキュリティグループIDsを検索し、エンドポイントのセキュリティグループの ID をコピーします。
4. ナビゲーションペインで、[Security groups (セキュリティグループ)] を選択します。
5. ターゲットに関連付けられているセキュリティグループのチェックボックスを選択し、[アクション]、[インバウンドルールの編集] を選択します。
6. Verified Access エンドポイントから発信するトラフィックを許可するセキュリティグループルールを追加するには、次の操作を行います。
   a. [ルールを追加] を選択します。
   b. [タイプ] で、[すべてのトラフィック]、または許可する特定のトラフィックを選択します。
   c. [ソース] で、[カスタム] を選択し、エンドポイントのセキュリティグループの ID を貼り付けます。
7. (オプション) トラフィックが Verified Access エンドポイントからのみ発信するようにするには、他のインバウンドセキュリティグループルールをすべて削除します。
8. [Save Rules] (ルールの保存) を選択します。

Verified Access エンドポイントの変更

Verified Access エンドポイントを変更するには、次の手順に従います。

Verified Access エンドポイントを変更するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. エンドポイントを選択します。
4. [アクション]、[Verified Access エンドポイントの変更] を選択します。
5. 必要に応じてエンドポイントの詳細を変更します。
6. [Verified Access エンドポイントの変更] を選択します。

Verified Access エンドポイントポリシーの変更

Verified Access エンドポイントのポリシーを変更するには、次の手順に従います。変更が有効になるまでに数分かかります。

Verified Access エンドポイントポリシーを変更するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. エンドポイントを選択します。
4. [アクション]、[Verified Access エンドポイントポリシーの変更] を選択します。
5. (オプション) 必要に応じてポリシーの有効化をオンまたはオフにします。
6. (オプション) ポリシー に、エンドポイントに適用する Verified Access ポリシーを入力します。
7. [Verified Access エンドポイントポリシーの変更]を選択します。

Verified Access エンドポイントの削除

不要になった VPC Access エンドポイントは、削除することができます。

Verified Access エンドポイントを削除するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. エンドポイントを選択します。

4. [アクション]、[Verified Access エンドポイントの削除] を選択します。
5. 確認を求められたら、**delete**と入力し、[削除] を選択します。

信頼プロバイダーから Verified Access に送信される信頼データ

信頼データは に送信されるデータです AWS Verified Access 信頼プロバイダーからの 。信頼データは、「ユーザークレーム」または「信頼コンテキスト」とも呼ばれます。データには通常、ユーザーまたはデバイスに関する情報が含まれます。信頼データの例には、ユーザー E メール、グループメンバーシップ、デバイスのオペレーティングシステムのバージョン、デバイスのセキュリティ状態などがあります。送信される情報は信頼プロバイダーによって異なるため、信頼プロバイダーのドキュメントで信頼データの完全かつ更新されたリストを参照する必要があります。

ただし、Verified Access のログ記録機能を使用すれば、信頼プロバイダーからどのようなトラストデータが送信されているかを確認することもできます。これは、アプリケーションへのアクセスを許可または拒否するポリシーを定義する場合に便利です。ログにトラストコンテキストを含める方法については、Verified Access 信頼コンテキストを有効または無効にする を参照してください。

このセクションには、ポリシーの記述を開始するのに役立つ信頼データのサンプルと例が含まれています。ここに記載されている情報は、例示を目的とするもので、正式なレファレンスではありません。

内容

Verified Access 信頼データのデフォルトコンテキスト

AWS Verified Access には、設定されている信頼プロバイダーに関係なく、すべての Cedar 評価に現在のHTTPリクエストに関するいくつかの要素がデフォルトで含まれています。ポリシーが評価されると、Verified Access は現在のHTTPリクエストに関するデータを の Cedar コンテキストに含めますcontext.http_request key。必要に応じて、データに対して評価を行うポリシーを作成できます。次のJSONスキーマは、評価に含まれるデータを示しています。

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

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

以下は、HTTPリクエストデータに対して を評価するポリシーの例です。

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

AWS IAM Identity Center Verified Access 信頼データのコンテキスト

ポリシーが評価されるときに、 AWS IAM Identity Center 信頼プロバイダーとして、 AWS Verified Access は、信頼プロバイダー設定で「ポリシーリファレンス名」として指定したキーの下の Cedar

コンテキストに信頼データを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。

> **Note**
>
> 信頼プロバイダーのコンテキストキーは、信頼プロバイダーの作成時に設定したポリシーレファレンス名から取得されます。たとえば、ポリシーレファレンスを「idp123」と設定した場合、コンテキストキーは「context.idp123」となります。ポリシーを作成する際は、正しいコンテキストキーを使用していることを確認してください。

次のJSONスキーマは、評価に含まれるデータを示しています。

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

以下は、 によって提供される信頼データに対して を評価するポリシーの例です。 AWS IAM Identity Center.

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

> **Note**
>
> グループ名は変更できるため、IAMIdentity Center はグループ ID を使用してグループを参照します。これにより、グループの名前を変更する際にポリシーステートメントが破られるのを防ぐことができます。

Verified Access 信頼データのサードパーティー信頼プロバイダーコンテキスト

このセクションでは、 に提供される信頼データについて説明します。 AWS Verified Access サードパーティーの信頼プロバイダーによる。

> **Note**
>
> 信頼プロバイダーのコンテキストキーは、信頼プロバイダーの作成時に設定したポリシーレファレンス名から取得されます。たとえば、ポリシーレファレンスを「idp123」と設定した場合、コンテキストキーは「context.idp123」となります。ポリシーを作成する際は、正しいコンテキストキーを使用していることを確認してください。

内容

ブラウザ拡張

デバイスの信頼コンテキストをアクセスポリシーに組み込む場合は、 AWS Verified Access ブラウザ拡張機能、または別のパートナーのブラウザ拡張機能。Verified Access は、現在 Google Chrome と Mozilla Firefox ブラウザをサポートしています。

現在、Jamf (macOS デバイスをサポート) 、 CrowdStrike (Windows 11 および Windows 10 デバイスをサポート) 、 JumpCloud (Windows と MacOS の両方をサポート) の 3 つのデバイス信頼プロバイダーをサポートしています。

- ポリシーで Jamf 信頼データを使用している場合、ユーザーは をダウンロードしてインストールする必要があります。 AWS Verified Access Chrome ウェブストアまたは Firefox アドオンサイトのブラウザ拡張機能。
- ポリシーでCrowdStrike信頼データを使用している場合は、まずユーザーが をインストールする必要があります。 AWS Verified Access ネイティブメッセージングホスト (直接ダウンロードリンク) 。このコンポーネントは、ユーザーのデバイスで実行されている CrowdStrike エージェントか

ら信頼データを取得するために必要です。次に、このコンポーネントをインストールした後、ユーザーは をインストールする必要があります。 AWS Verified Access Chrome ウェブストアまたは Firefox アドオンサイトのブラウザ拡張機能。

- を使用している場合JumpCloud、ユーザーは Chrome ウェブストアまたは Firefox アドオンサイトの JumpCloud ブラウザ拡張機能をデバイスにインストールする必要があります。

Jamf

Jamf はサードパーティー信頼プロバイダーです。ポリシーが評価される際に、Jamf を信頼プロバイダーとして定義すると、Verified Access は、信頼プロバイダー設定で「ポリシーレファレンス名」として指定するキーの下の Cedar コンテキスト内のトラストデータを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。次のJSONスキーマは、評価に含まれるデータを示しています。

Verified Access で Jamf を使用する方法の詳細については、Jamf ウェブサイトの「Integrating AWS Verified Access with Jamf Device Identity」を参照してください。

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

以下は、Jamf が提供するトラストデータに対して評価を行うポリシーの例です。

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

Cedar には、Jamf のリスクスコアなどの列挙型を使用する際に役立つ便利な `.contains()` 機能があります。

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

CrowdStrike

CrowdStrike はサードパーティーの信頼プロバイダーです。ポリシーが評価されると、 を信頼プロバイダー CrowdStrike として定義すると、Verified Access は、信頼プロバイダー設定で「ポリシー

参照名」として指定したキーの下の Cedar コンテキストに信頼データを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。次のJSONスキーマは、評価に含まれるデータを示しています。

Verified Access CrowdStrike で を使用する方法の詳細については、「 CrowdStrike と によるプライベートアプリケーションの保護」を参照してください。 AWS Verified Access GitHub ウェブサイトの 。

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

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

以下は、 によって提供される信頼データに対して を評価するポリシーの例です CrowdStrike。

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

JumpCloud

JumpCloud はサードパーティーの信頼プロバイダーです。ポリシーが評価されると、信頼プロバイダー JumpCloud として を定義すると、Verified Access は、信頼プロバイダー設定で「ポリシー

リファレンス名」として指定したキーの Cedar コンテキストに信頼データを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。次のJSONスキーマは、評価に含まれるデータを示しています。

JumpCloud で を使用する方法の詳細については、「」を参照してください。 AWS Verified Access、「Integrating JumpCloud and AWS JumpCloud ウェブサイトの 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"
    }
  }
}
```

以下は、 によって提供される信頼コンテキストに対して を評価するポリシーの例です JumpCloud。

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

Verified Access でのユーザークレームの受け渡しと署名の検証

の後 AWS Verified Access インスタンスはユーザーを正常に認証し、IdP から受信したユーザークレームを Verified Access エンドポイントに送信します。ユーザークレームは、アプリケーションが署名を検証し、クレームが Verified Access によって送信されたことを確認することができるように署名されます。このプロセス中に、次のHTTPヘッダーが追加されます。

`x-amzn-ava-user-context`

このヘッダーには、JSONウェブトークン (JWT) 形式のユーザークレームが含まれます。JWT 形式には、base64 でURLエンコードされたヘッダー、ペイロード、署名が含まれます。Verified Access は ES384 (ECDSA SHA-384 ハッシュアルゴリズムを使用する署名アルゴリズム) を使用してJWT署名を生成します。

アプリケーションはこれらのクレームをパーソナライズやその他のユーザー固有のエクスペリエンスに使用できます。アプリケーションデベロッパーは、使用前に ID プロバイダーから提供される各クレームの一意性と検証のレベルについて十分に理解しておく必要があります。一般に、sub クレームは、特定のユーザーを識別する最良の方法です。

内容

- パブリックキー
- 例: 取得とデコード JWT

例: OIDC ユーザークレームJWTの署名

次の例は、OIDCユーザークレームのヘッダーとペイロードが JWT 形式でどのように表示されるかを示しています。

ヘッダーの例：

```
{
   "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)
}
```

ペイロードの例：

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

例: IAM Identity Center ユーザークレームJWTの署名

次の例は、IAMIdentity Center ユーザークレームのヘッダーとペイロードが JWT 形式でどのように表示されるかを示しています。

> **Note**
>
> IAM Identity Center では、ユーザー情報のみがクレームに含まれます。

ヘッダーの例：

```
{
   "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)
}
```

ペイロードの例：

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

パブリックキー

Verified Access インスタンスはユーザークレームを暗号化しないため、 を使用するように Verified Access エンドポイントを設定することをお勧めしますHTTPS。を使用するように Verified Access エンドポイントを設定する場合はHTTP、セキュリティグループを使用してエンドポイントへのトラフィックを必ず制限してください。

セキュリティを確保するには、クレームに基づいて認証を行う前に署名を検証し、 JWTヘッダーの `signer`フィールドに期待される Verified Access インスタンス が含まれていることを確認する必要がありますARN。

パブリックキーを取得するには、 JWTヘッダーからキー ID を取得し、それを使用してエンドポイントからパブリックキーを検索します。

各 のエンドポイント AWS リージョン は次のとおりです。

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

例: 取得とデコード JWT

次のコードは、Python 3.9 でキー ID、公開キー、およびペイロードを取得する方法を示しています。

```
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'])
```

Verified Access ポリシー

AWS Verified Access ポリシーでは、 でホストされているアプリケーションにアクセスするためのルールを定義できます。 AWS。 これらは Cedar、 AWS ポリシー言語。Cedar を使用すると、Verified Access で使用するように設定する ID またはデバイスベースの信頼プロバイダーから送信されたトラストデータに基づいて評価されるポリシーを作成できます。

Cedar ポリシー言語の詳細については、「Cedar リファレンスガイド」をご覧ください。

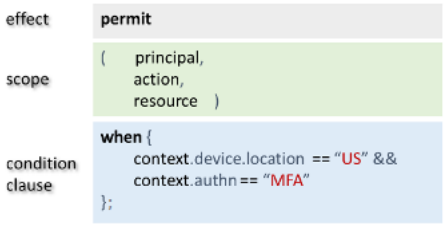
Verified Access グループを作成する時、または Verified Access エンドポイントを作成する時、オプションとして Verified Access ポリシーを定義できます。Verified Access ポリシーを定義しなくてもグループまたはエンドポイントを作成できますが、ポリシーを定義するまですべてのアクセス要求はブロックされます。または、作成後に既存の Verified Access グループまたはエンドポイントでポリシーを追加または変更できます。

このセクションでは、Verified Access ポリシーの構造、内容、定義方法について説明し、例をいくつか示します。

内容

Verified Access ポリシーステートメント構造

このセクションでは、 AWS Verified Access ポリシーステートメントとその評価方法。ひとつの Verified Access ポリシーに複数のステートメントを設定できます。Verified Access ポリシーの構造は、以下の図の通りです。



ポリシーには、次の部分が含まれます。

- 効果 — ポリシーステートメントが `permit` (Allow) か `forbid` (Deny) かを指定します。
- 対象範囲 — 効果を適用するプリンシパル、アクション、リソースを指定します。特定のプリンシパル、アクション、またはリソースを特定しないことで、Cedar の対象範囲を未定義のままにしておくことができます (前の例を参照)。この場合、ポリシーはすべてのプリンシパル、アクション、リソースに適用されます。
- 条件節 — 効果が適用されるコンテキストを指定します。

> **Important**
>
> Verified Access では、条件節に含まれるトラストデータを参照することでポリシーが完全に表現されます。ポリシーの対象範囲は、常に未定義のままにしておく必要があります。その後、条件節に含まれる ID とデバイスのトラストコンテキストを使用してアクセスを指定できます。

簡単なポリシーの例

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

前の例では、&& 演算子を使用して 1 つのポリシーステートメントに複数の条件節を使用できることにご注意ください。Cedar のポリシー言語を使うと、カスタムできめ細かな広範囲にわたるポリシーステートメントを作成する表現力が得られます。その他の例については、「Verified Access ポリシーの例」を参照してください。

Verified Access ポリシーの評価

ポリシードキュメントは 1 つ以上のポリシーステートメント (`permit` または `forbid` ステートメント) のセットです。ポリシーは、条件節 (when ステートメント) が true である場合に適用されます。ポリシードキュメントがアクセスを許可するには、ドキュメント内の少なくとも 1 つの許可ポリシーが適用されている必要があり、禁止ポリシーを適用することはできません。許可ポリシーが適用されない場合や、1 つ以上の禁止ポリシーが適用されている場合、ポリシードキュメントはアクセスを拒否します。Verified Access グループと Verified Access エンドポイントの両方に定義されたポリシードキュメントがある場合、両方のドキュメントがアクセスを許可する必要があります。Verified Access エンドポイントのポリシードキュメントを定義していない場合は、アクセスを許可する必要があるのは Verified Access グループポリシーのみです。

> **Note**
>
> AWS Verified Access はポリシーの作成時に構文を検証しますが、条件句に入力したデータは検証しません。

Verified Access ポリシーの組み込み演算子

のコンテキストを作成する場合 AWS Verified Access 「」で説明されているように、さまざまな条件を使用する ポリシーではVerified Access ポリシーステートメント構造、 &&演算子を使用して条件を追加できます。ポリシー条件にさらに表現力を加えるために使用できるビルトイン演算子は他にも多数あります。参照用にすべてのビルトイン演算子を下表に示します。

演算子	タイプとオーバーロード	説明
!	Boolean → Boolean	論理否定。
==	any → any	等価。タイプが一致しなくても、いずれかのタイプの引数で機能します。異なるタイプの値が互いに等しくなることはありません。
!=	any → any	不等価。等価の正反対 (上記参照)。

演算子	タイプとオーバーロード	説明
<	(long, long) → Boolean	Long integer less-than.
<=	(long, long) → Boolean	整数 less-than-or-equal-to。
>	(long, long) → Boolean	Long integer greater-than.
>=	(long, long) → Boolean	整数 greater-than-or-equal-to。
in	(entity, entity) → Boolean	階層メンバーシップ (再帰形：A の A は常にトゥルー)。
	(entity, set(entity)) → Boolean	階層メンバーシップ：(A and B) || (A in C) || であれば A in [B, C, ...] はトゥルー... セットに non-entity が含まれている場合はエラーになります。
&&	(Boolean, Boolean) → Boolean	Logical and (short-circuiting).
||	(Boolean, Boolean) → Boolean	Logical or (short-circuiting).
.exists()	entity → Boolean	Entity existence.
has	(entity, attribute) → Boolean	中置演算子。 `e has f` レコードまたはエンティティに e 属性 f へのバインディングがあるかどうかをテストします。e が存在しないか、e が存在しても属性 f を持たない場合は `false` を返します。属性は識別子または文字列リテラルとして表現できます。

演算子	タイプとオーバーロード	説明
like	(string, string) → Boolean	中置演算子。`t like p` テキスト `t` がパターン `p` と一致するかどうかを確認します。パターンには、0 個以上の文字と一致するワイルドカード文字 `*` が含まれる場合があります。`t` のリテラルスター文字と一致させるには、`p` で特殊なエスケープ文字シーケンス `\*` を使用できます。
.contains()	(set, any) → Boolean	メンバーシップ (B が A の要素かどうか) を設定します。
.containsAll()	(set, set) → Boolean	セット A にセット B のすべての要素が含まれているかどうかをテストします。
.containsAny()	(set, set) → Boolean	セット A にセット B の要素のいずれかが含まれているかどうかをテストします。

Verified Access ポリシーのコメント

にコメントステートメントを含めることができます。 AWS Verified Access ポリシー。コメントは、`//` で始まり改行で終わる行として定義されます。

次の例は、ポリシー内のコメントステートメントを示しています。

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

```
};
```

Verified Access ポリシーロジックのショートサーキット

を書き込むことができます。 AWS Verified Access 特定のコンテキストに存在する場合と存在しない場合があるデータを評価する ポリシー。存在しないコンテキスト内のデータを参照すると、意図に関係なく、Cedar はエラーを作成し、ポリシーでアクセスが拒否されるよう評価します。例えば、fake_provider と bogus_key はこのコンテキストでは存在しないため、結果的に拒否されます。

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

このような状況を回避するには、 has演算子を使用してキーが存在するかどうかを確認できます。has 演算子が false を返すと、連鎖ステートメントのさらなる評価は停止し、Cedar は存在しない項目の参照を試みることによるエラーを生成しません。

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

これは、2 つの異なる信頼プロバイダーを参照するポリシーを指定する場合に最も有用です。

```
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)
    )
```

```
  )
};
```

Verified Access ポリシーの例

例 1: IAM Identity Center のポリシーの作成

> **Note**
>
> グループ名は変更できるため、IAMIdentity Center はグループ ID を使用してグループを参照します。これにより、グループの名前を変更する際にポリシーステートメントが破られるのを防ぐことができます。

以下のポリシー例では、ユーザーが `finance` グループ (グループ ID は `c242c5b0-6081-1845-6fa8-6e0d9513c107`) に属し、検証済みメールアドレスを持っている場合にのみアクセスを許可します。

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

例 1b: IAM Identity Center のポリシーステートメントに条件を追加する

以下のポリシー例では、ユーザーが `finance` グループ (グループ ID は `c242c5b0-6081-1845-6fa8-6e0d9513c107`) に属し、検証済みメールアドレスを持っていて、Jamf デバイスリスクスコアが `LOW` の場合にのみアクセスを許可します。

```
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"
};
```

例 2: サードパーティーOIDCプロバイダーの同じポリシー

次のポリシー例では、ユーザーが「財務」グループのユーザーで、検証済みの E メールアドレスがあり、Jamf デバイスリスクスコアが の場合にのみアクセスを許可しますLOW。

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

例 3: の使用 CrowdStrike

次のポリシー例では、全体評価のスコアが 50 を超えるとアクセスが許可されます。

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

例 4：特殊文字の使用

次の例では、コンテキストプロパティがポリシー言語の予約文字である：(セミコロン) を使用している場合のポリシーの記述方法を示しています。

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

例 5：特定の IP アドレスの許可

以下は、特定の IP アドレスのみを許可するポリシーの例です。

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

例 5a：特定の IP アドレスのブロック

以下は、特定の IP アドレスをブロックするポリシーの例です。

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

Verified Access ポリシーアシスタント

Verified Access ポリシーアシスタントは、ポリシーのテストと開発に使用できる Verified Access コンソールに含まれるツールです。エンドポイントポリシー、グループポリシー、およびトラストコンテキストが 1 つの画面に表示され、そこでポリシーをテストしたり編集したりできます。

トラストコンテキストの形式は信頼プロバイダーごとに異なり、Verified Access 管理者は特定の信頼プロバイダーの使用する正確な形式を知らない場合があります。そのため、テストや開発の目的で、信頼コンテキストとグループポリシーとエンドポイントポリシーの両方を 1 か所で確認できると非常に便利です。

以下のセクションでは、ポリシーエディターを使用するうえでの基本事項を説明します。

タスク

- ステップ 1: リソースを指定する
- ステップ 2: ポリシーをテストおよび編集する
- ステップ 3: 変更を確認して適用する

ステップ 1: リソースを指定する

ポリシーアシスタントの最初のページで、使用する Verified Access エンドポイントを指定します。また、ユーザー (E メールアドレスで識別) を指定し、オプションでユーザーの名前および/またはデバイス ID を指定します。デフォルトでは、指定したユーザーの Verified Access ログから最新の認証決定が抽出されます。オプションで、最新の許可または拒否の決定を具体的に選択できます。

最後に、トラストコンテキスト、認証決定、エンドポイントポリシー、およびグループポリシーがすべて次の画面に表示されます。

ポリシーアシスタントを開いてリソースを指定するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで [Verified Access インスタンス] を選択し、操作するインスタンスの [Verified Access インスタンス ID] をクリックします。

3. [ポリシーアシスタントを起動] を選択します。
4. [ユーザーの E メールアドレス] で、ユーザーのメールアドレスを入力します。
5. [Verified Access エンドポイント] では、ポリシーを編集してテストするエンドポイントを選択します。
6. (オプション) [名前] には、ユーザーの名前を入力します。
7. (オプション) [デバイス識別子] に、一意のデバイス識別子を指定します。
8. (オプション)[認証結果] では、使用する最新の認証結果の種類を選択します。デフォルトでは、最新の認証結果が使用されます。
9. [次へ] を選択します。

ステップ 2: ポリシーをテストおよび編集する

このページには、次で作業するための情報が表示されます。

- 信頼プロバイダーがユーザーと (オプションで) 前のステップで指定したデバイスのために送信したトラストコンテキスト。
- 前のステップで指定された Verified Access エンドポイントの Cedar ポリシー。
- エンドポイントが属する Verified Access グループの Cedar ポリシー。

Verified Access エンドポイントとグループの Cedar ポリシーはこのページで編集できますが、トラストコンテキストは静的です。このページを使用して、Cedar ポリシーと一緒にトラストコンテキストを表示できるようになりました。

[ポリシーをテスト] ボタンを選択して、トラストコンテキストに対し、ポリシーをテストすると、認証結果が画面に表示されます。必要に応じてこのプロセスを繰り返すことで、ポリシーを編集して変更を再テストできます。

ポリシーの変更に問題がなければ、[次へ] を選択してポリシーアシスタントの次の画面に進みます。

ステップ 3: 変更を確認して適用する

ポリシーアシスタントの最後のページでは、ポリシーに加えた変更が強調表示され、簡単に確認できます。これで、変更内容を最後に確認し、[変更を適用]] を選択して変更を確定できます。

また、[前へ] を選択して前のページに戻るか、[キャンセル] を選択してポリシーアシスタントを完全にキャンセルすることもできます。

Verfied Access のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、 AWS とユーザーの間で共有される責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ — AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任を担います AWS クラウド。また、 は、お客様が安全に使用できるサービス AWS も提供します。コンプライアンスAWS プログラムコンプライアンスプログラム の一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。 AWS Verified Access に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウドのセキュリティ — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、Verified Access を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために Verified Access を設定する方法を示します。また、Verified Access リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

内容

Verified Access でのデータ保護

- AWS 責任共有モデル 、 でのデータ保護に適用されます。 AWS Verified Access。このモデルで説明されているように、 AWS は、すべての を実行するグローバルインフラストラクチャを保護する責任があります。 AWS クラウド。 お客様は、このインフラストラクチャでホストされているコン

テンツの制御を維持する責任があります。また、 のセキュリティ設定と管理タスクについても責任を負います。 AWS のサービス 使用する 。データプライバシーの詳細については、「データプライバシーFAQ」を参照してください。欧州でのデータ保護の詳細については、「」を参照してください。 AWS の責任共有モデルとGDPRブログ記事 AWS セキュリティブログ 。

データ保護の目的で、 を保護することをお勧めします。 AWS アカウント 認証情報と を使用して個々のユーザーをセットアップする AWS IAM Identity Center または AWS Identity and Access Management （ IAM ）。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して と通信する AWS リソースの使用料金を見積もることができます。1TLS.2 が必要で、1.3 TLS をお勧めします。
- で APIとユーザーアクティビティのログ記録を設定する AWS CloudTrail。 CloudTrail 証跡を使用してキャプチャする方法については、「」を参照してください。 AWS アクティビティ、「」の「証 CloudTrail 跡の使用」を参照してください。 AWS CloudTrail ユーザーガイド 。
- 使用アイテム AWS 暗号化ソリューションと 内のすべてのデフォルトのセキュリティコントロール AWS のサービス.
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- アクセス時に FIPS 140-3 検証済みの暗号化モジュールが必要な場合 AWS コマンドラインインターフェイスまたは を介してAPI、FIPSエンドポイントを使用します。利用可能なFIPSエンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これは、Verified Access またはその他の を使用する場合も同様です。 AWS のサービス コンソール、API、 AWS CLI、または AWS SDKs。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。URL を外部サーバーに提供する場合は、そのサーバーへのリクエストを検証URLするために認証情報を に含めないことを強くお勧めします。

転送中の暗号化

Verified Access は、Transport Layer Security (TLS) 1.2 以降を使用して、インターネット経由でエンドユーザーから Verified Access エンドポイントに転送されるすべてのデータを暗号化します。

ネットワーク間トラフィックのプライバシー

Verified Access を設定して、 内の特定のリソースへのアクセスを制限できますVPC。ユーザーベースの認証の場合、 エンドポイントにアクセスするユーザーグループに基づいて、ネットワークの一部へのアクセスを制限することもできます。詳細については、「Verified Access ポリシー」を参照してください。

の保管中のデータ暗号化 AWS Verified Access

AWS Verified Access は、デフォルトで を使用して保管中のデータを暗号化します。 AWS 所有KMSキー。デフォルトで保管中のデータを暗号化すると、機密データの保護に伴う運用上のオーバーヘッドや複雑さを軽減できます。同時に、暗号化のコンプライアンスと規制の厳格な要件を満たす、安全なアプリケーションを構築することもできます。以下のセクションでは、Verified Access が保管中のデータ暗号化にKMSキーを使用する方法の詳細を説明します。

内容

Verified Access とKMSキー

AWS 所有キー

Verified Access はKMSキーを使用して、個人を特定できる情報 () を自動的に暗号化しますPII。これはデフォルトで発生し、AWS所有キーの使用を自分で表示、管理、使用、または監査することはできません。ただし、データを暗号化するキーを保護するためのアクションの実施やプログラムの変更を行う必要はありません。詳細については、「」を参照してくださいAWS の 所有キー AWS Key Management Service デベロッパーガイド 。

この暗号化レイヤーを無効にしたり、代替の暗号化タイプを選択したりすることはできませんが、既存の に 2 つ目の暗号化レイヤーを追加できます。 AWS Verified Access リソースの作成時にカスタマーマネージドキーを選択して、 が所有する暗号化キー。

カスタマーマネージドキー

Verified Access では、お客様が作成して管理する対称カスタマーマネージドキーを使用して、既存のデフォルトの暗号化に 2 番目の暗号化レイヤーを追加することができます。この暗号化レイヤーはユーザーが完全に制御できるため、次のようなタスクを実行できます。

- キーポリシーの策定と維持
- IAM ポリシーと許可の確立と維持
- キーポリシーの有効化と無効化
- キー暗号化マテリアルのローテーション
- タグの追加
- キーエイリアスの作成
- キー削除のスケジュール設定

詳細については、「」の「カスタマーマネージドキー」を参照してください。 AWS Key Management Service デベロッパーガイド 。

> **Note**
>
> Verified Access は、 を使用して保管時の暗号化を自動的に有効にします。 AWS 個人を特定できるデータを無料で保護するための 所有キー。
> ただし、 AWS KMS カスタマーマネージドキーを使用する場合、 料金が適用されます。料金の詳細については、「」を参照してください。 AWS Key Management Service の料金

個人を特定できる情報

次の表は、Verified Access が使用する個人を特定できる情報 (PII) とその暗号化方法をまとめたものです。

データ型	AWS 所有キーの暗号化	カスタマーマネージドキーの暗号化 (オプション)
`Trust provider (user-type)`	有効	有効

データ型	AWS 所有キーの暗号化	カスタマーマネージドキーの暗号化 (オプション)
ユーザータイプの信頼プロバイダーには AuthorizationEndpoint、、 UserInfoEndpoint ClientId、 ClientSecret、 などのOIDCオプションが含まれており、これらは と見なされますPII。		
`Trust provider (device-type)` デバイスタイプの信頼プロバイダーには TenantId、 と見なされる が含まれていますPII。	有効	有効
`Group policy` Verified Access グループの作成または変更時に提供されます。アクセス要求を承認するためのルールが含まれています。ユーザー名や E メールアドレスPIIなどが含まれる場合があります。	有効	有効
`Endpoint policy` Verified Access エンドポイントの作成または変更時に提供されます。アクセス要求を承認するためのルールが含まれています。ユーザー名や E メールアドレスPIIなどが含まれる場合があります。	有効	有効

その方法は? AWS Verified Access は で許可を使用します AWS KMS

Verified Access では、カスタマーマネージドキーを使用するためのグラントが必要です。

カスタマーマネージドキーで暗号化された Verified Access リソースを作成すると、Verified Access は にCreateGrantリクエストを送信してユーザーに代わって許可を作成します。 AWS KMS。 の許可 AWS KMS は、Verified Access にアカウントのカスタマーマネージドキーへのアクセスを許可するために使用されます。

Verified Access では、以下の内部オペレーションでカスタマーマネージドキーを使用するためにグラントが必要です。

- Decrypt リクエストを に送信する AWS KMS は、暗号化されたデータキーを復号化して、データの復号化に使用できます。
- へのRetireGrantリクエストの送信 AWS KMS 権限を削除する場合。

任意のタイミングで、許可に対するアクセス権を取り消したり、カスタマーマネージドキーに対するサービスからのアクセス権を削除したりできます。これを行うと、Verified Access はカスタマーマネージドキーによって暗号化されたすべてのデータにアクセスできなくなり、そのデータに依存しているオペレーションが影響を受けます。

Verified Access でカスタマーマネージドキーを使用する

対称カスタマーマネージドキーは、 を使用して作成できます。 AWS Management Console、または AWS KMS APIs。「」の「対称カスタマーマネージドキーの作成」の手順に従います。 AWS Key Management Service デベロッパーガイド 。

キーポリシー

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが 1 つだけ必要です。このポリシーには、そのキーを使用できるユーザーとその使用方法を決定するステートメントが含まれています。カスタマーマネージドキーを作成する際に、キーポリシーを指定することができます。詳細については、「」の「カスタマーマネージドキーへのアクセスの管理」を参照してください。 AWS Key Management Service デベロッパーガイド 。

Verified Access リソースでカスタマーマネージドキーを使用するには、キーポリシーで次のAPIオペレーションを許可する必要があります。

- kms:CreateGrant - カスタマーマネージドキーに許可を追加します。指定されたKMSキーへのアクセスを制御する権限を付与します。これにより、Verified Access が必要とする許可オペレーションへのアクセスを許可します。Grants の使用の詳細については、「」を参照してください。AWS Key Management Service デベロッパーガイド 。

  これにより、Verified Access が以下を実行できるようになります。

  - GenerateDataKeyWithoutPlainText を呼び出して暗号化されたデータキーを生成して保存します。データキーは暗号化にすぐには使用されないからです。
  - Decrypt を呼び出して、保存された暗号化データキーを使用して暗号化データにアクセスします。
  - 廃止するプリンシパルを設定して、サービスがRetireGrantを実行できるようにします。
- kms:DescribeKey — カスタマーマネージドキーの詳細を提供し、Verified Access がキーを検証できるようにします。
- kms:GenerateDataKey — Verified Access がキーを使用してデータを暗号化できるようにします。
- kms:Decrypt — Verified Access が暗号化されたデータを復号できるようにします。

以下は、Verified Access で使用できるキーポリシーの例です。

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

ポリシー でのアクセス許可の指定の詳細については、「」を参照してください。 AWS Key Management Service デベロッパーガイド 。

キーアクセスのトラブルシューティングの詳細については、「」を参照してください。 AWS Key Management Service デベロッパーガイド 。

Verified Access リソースのカスタマーマネージドキーを指定する

カスタマーマネージドキーを指定して、以下のリソースに 2 番目の暗号化レイヤを提供できます。

- Verified Access グループ
- Verified Access エンドポイント
- Verified Access 信頼プロバイダー

を使用してこれらのリソースを作成する場合 AWS Management Consoleでは、「追加の暗号化 -- オプション」セクションでカスタマーマネージドキーを指定できます。プロセス中に、暗号化設定のカスタマイズ (アドバンスト) チェックボックスを選択し、 AWS KMS 使用するキー ID。これは、既存のリソースを変更するとき、または を使用して行うこともできます。 AWS CLI.

> **Note**
>
> 上記のリソースのいずれかに暗号化を追加するために使用されるカスタマーマネージドキーが失われると、リソースの設定値にアクセスできなくなります。ただし、 を使用してリソースを変更できます。 AWS Management Console または AWS CLI、新しいカスタマーマネージドキーを適用し、設定値をリセットします。

AWS Verified Access 暗号化コンテキスト

暗号化コンテキストは、データに関する追加のコンテキスト情報が含まれたキーバリューペアのオプションのセットです。 AWS KMS は、認証された暗号化 をサポートするために、暗号化コンテキストを追加の認証データとして使用します。データの暗号化リクエストに暗号化コンテキストを含めると、 AWS KMS は、暗号化コンテキストを暗号化されたデータにバインドします。データを復号化するには、そのリクエストに (暗号化時と) 同じ暗号化コンテキストを含めます。

AWS Verified Access 暗号化コンテキスト

Verified Access は、すべての で同じ暗号化コンテキストを使用します。 AWS KMS 暗号化オペレーション。キーは `aws:verified-access:arn`で、値はリソース Amazon リソースネーム () です ARN。Verified Access リソースの暗号化コンテキストは次のとおりです。

Verified Access 信頼プロバイダー

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

Verified Access グループ

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

Verified Access エンドポイント

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

許可またはポリシーでの暗号化コンテキストの使用の詳細については、「」の「暗号化コンテキスト」を参照してください。 AWS Key Management Service デベロッパーガイド 。

の暗号化キーのモニタリング AWS Verified Access

でカスタマーマネージドKMSキーを使用する場合 AWS Verified Access リソース、 AWS CloudTrail Verified Access が に送信するリクエストを追跡するには AWS KMS.

以下の例を示します。 AWS CloudTrail 、CreateGrant、、RetireGrantDecrypt、DescribeKeyおよび の イベントGenerateDataKey。カスタマーマネージドKMSキーによって暗号化されたデータにアクセスするために Verified Access によって呼び出されるKMSオペレーションをモニタリングします。

CreateGrant

カスタマーマネージドキーを使用してリソースを暗号化すると、Verified Access はユーザーに代わって のキーにアクセスするCreateGrantリクエストを送信します。 AWS アカウント。Verified Access が作成するグラントは、カスタマーマネージドキーに関連付けられているリソースに固有のものです。

以下のイベント例は、CreateGrant オペレーションを記録したものです。

```
{
    "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 では、リソースを削除するときに、RetireGrant オペレーションを使用してグラントを削除します。

以下のイベント例は、RetireGrant オペレーションを記録したものです。

```
{
    "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 は、保存されている暗号化データキーを使用して暗号化されたデータにアクセスするために Decrypt オペレーションを呼び出します。

以下のイベント例は、Decrypt オペレーションを記録したものです。


```
{
    "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 は DescribeKey オペレーションを使用して、リソースに関連付けられているカスタマーマネージドキーがアカウントおよびリージョンに存在するかどうかを確認します。

以下のイベント例は、DescribeKey オペレーションを記録したものです。

```
{
    "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

以下のイベント例は、GenerateDataKey オペレーションを記録したものです。

```
{
    "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"
}
```

Verified Access のID およびアクセス管理

AWS Identity and Access Management （ IAM) は AWS のサービス 管理者が へのアクセスを安全に制御するのに役立ちます。 AWS リソースの使用料金を見積もることができます。IAM 管理者は、誰を認証 (サインイン) し、誰に Verified Access リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は AWS のサービス 追加料金なしで使用できます。

トピック

対象者

の使用方法 AWS Identity and Access Management （ IAM) は、Verified Access で行う作業によって異なります。

サービスユーザー – Verified Access サービスを使用してジョブを実行する場合は、必要な認証情報とアクセス許可を管理者が提供します。作業を実行するためにさらに多くの Verified Access の機能を使用するとき、追加の許可が必要になる場合があります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。Veriied Access の機能にアクセスできない場合は、「 Verified Access アイデンティティとアクセスに関するトラブルシューティング」を参照してください。

サービス管理者 – 社内の Verified Access リソースを担当している場合は、通常、Verified Access に完全にアクセスすることができます。サービスのユーザーがどの Verified Access 機能やリソースにアクセスするかを決めるのは管理者の仕事です。次に、サービスユーザーのアクセス許可を変更するリクエストをIAM管理者に送信する必要があります。このページの情報を確認して、 の基本概念を理解してくださいIAM。会社で Verified Access IAMを使用する方法の詳細については、「」を参照してくださいVerified Access と の連携方法 IAM。

IAM 管理者 – IAM管理者は、Verified Access へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。で使用できる Verified Access アイデンティティベースのポリシーの例を表示するにはIAM、「」を参照してくださいVerified Access のアイデンティティベースポリシーの例。

アイデンティティを使用した認証

認証は、 にサインインする方法です。 AWS ID 認証情報を使用する。認証されている必要があります (にサインインします AWSとしての) AWS アカウントのルートユーザー、 IAM ユーザーとして、または IAMロールを引き受ける方法。

にサインインできます。 AWS ID ソースを通じて提供される認証情報を使用して、フェデレーティッド ID として を指定します。 AWS IAM Identity Center （ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーティッド ID の例です。フェデレーティッド ID としてサインインすると、管理者は以前に IAMロールを使用して ID フェデレーションをセットアップしていました。にアクセスする場合 AWS フェデレーションを使用すると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 にサインインできます。 AWS Management Console または AWS アクセスポータル。へのサインインの詳細については、「」を参照してください。 AWS「 にサインインする方法」を参照してください。 AWS アカウント ()AWS サインイン ユーザーガイド 。

アクセスする場合 AWS プログラムにより、 AWS は、 認証情報を使用してリクエストに暗号で署名するためのソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) を提供します。を使用しない場合 AWS ツール、リクエストには自分で署名する必要があります。推奨される方法を使用してリクエストに自分で署名する方法の詳細については、「署名」を参照してください。 AWS API IAMユーザーガイドの リクエスト。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、 などです AWS では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用することをお勧めします。詳細については、「」の「多要素認証」を参照してください。 AWS IAM Identity Center ユーザーガイドと での多要素認証 (MFA) の使用 AWS「」(IAM ユーザーガイド) を参照してください。

AWS アカウント ルートユーザー

を作成する場合 AWS アカウントでは、すべての への完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。 AWS のサービス アカウントの および リソース。この ID は と呼ばれます。 AWS アカウント root ユーザー と には、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用

しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「 IAMユーザーガイド」の「ルートユーザーの認証情報を必要とするタスク」を参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、ID プロバイダーとのフェデレーションを使用して にアクセスすることを要求します。 AWS のサービス一時的な認証情報を使用する。

フェデレーティッド ID は、エンタープライズユーザーディレクトリのユーザー、ウェブ ID プロバイダー、 AWS Directory Service、Identity Center ディレクトリ、または にアクセスする任意のユーザー AWS のサービス ID ソースを通じて提供される認証情報を使用する。フェデレーティッド ID アクセスの場合 AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

一元的なアクセス管理を行うには、 を使用することをお勧めします。 AWS IAM Identity Center。Identity Center でユーザーとグループを作成するか、独自の IAM ID ソース内のユーザーとグループのセットに接続して同期し、すべての で使用できます。 AWS アカウント および アプリケーション。IAM Identity Center の詳細については、「」のIAM「Identity Center とは」を参照してください。 AWS IAM Identity Center ユーザーガイド 。

IAM ユーザーとグループ

IAM ユーザーは 内のアイデンティティです。 AWS アカウント 1 人のユーザーまたはアプリケーションに対して特定のアクセス許可を持つ 。可能な場合は、パスワードやアクセスキーなどの長期的な認証情報を持つIAMユーザーを作成するのではなく、一時的な認証情報を使用することをお勧めします。ただし、IAMユーザーとの長期的な認証情報を必要とする特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「 ユーザーガイド」の「長期的な認証情報を必要とするユースケースでアクセスキーを定期的にローテーションするIAM」を参照してください。

IAM グループは、IAMユーザーのコレクションを指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、 という名前のグループIAMAdminsを作成し、そのグループにIAMリソースを管理するアクセス許可を付与できます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユー

ザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。詳細については、「 ユーザーガイド」のIAM「 (ロールではなく) ユーザーを作成する場合IAM」を参照してください。

IAM ロール

IAM ロールは 内のアイデンティティです。 AWS アカウント 特定のアクセス許可を持つ 。これは IAM ユーザーと似ていますが、特定のユーザーに関連付けられていません。で一時的に IAMロールを引き受けることができます。 AWS Management Console ロールを切り替えます。を呼び出すことでロールを引き受けることができます。 AWS CLI または AWS API オペレーション、またはカスタム の使用URL。ロールの使用方法の詳細については、「 ユーザーガイド」のIAM「ロールの使用IAM」を参照してください。

IAM 一時的な認証情報を持つ ロールは、以下の状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、「 ユーザーガイド」の「サードパーティー ID プロバイダーのロールの作成IAM」を参照してください。IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証後にアクセスできる内容を制御するために、IAMIdentity Center はアクセス許可セットを のロールに関連付けますIAM。アクセス許可セットの詳細については、「」の「アクセス許可セット」を参照してください。 AWS IAM Identity Center ユーザーガイド 。
- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、 IAMロールを引き受けて、特定のタスクに対して異なるアクセス許可を一時的に引き受けることができます。
- クロスアカウントアクセス – IAMロールを使用して、別のアカウントのユーザー (信頼されたプリンシパル) がアカウントのリソースにアクセスすることを許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、 AWS のサービスでは、ポリシーをリソースに直接アタッチできます (ロールをプロキシとして使用する代わりに) 。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、「 ユーザーガイド」の「 でのクロスアカウントリソースアクセスIAMIAM」を参照してください。
- クロスサービスアクセス – 一部 AWS のサービス 他の の機能を使用する AWS のサービス。 例えば、サービスで呼び出しを行うと、そのサービスが Amazon でアプリケーションを実行EC2したり、Amazon S3 にオブジェクトを保存したりするのが一般的です。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。

- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を使用します。 AWS のサービス、 リクエストとの組み合わせ AWS のサービス ダウンストリームサービスにリクエストを行う 。FAS リクエストは、サービスが他の とのやり取りを必要とするリクエストを受け取った場合にのみ行われます。 AWS のサービス または完了するリソース。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
- サービスロール – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受ける IAMロールです。IAM 管理者は、 内からサービスロールを作成、変更、削除できますIAM。詳細については、「 にアクセス許可を委任するロールの作成」を参照してください。 AWS のサービス「」(IAM ユーザーガイド) を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です。 AWS のサービス。 このサービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールが に表示されます。 AWS アカウント と は サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

- Amazon で実行されているアプリケーション EC2 – IAMロールを使用して、EC2インスタンスで実行され、 を作成しているアプリケーションの一時的な認証情報を管理できます。 AWS CLI または AWS API リクエスト。これは、EC2インスタンス内にアクセスキーを保存するよりも望ましいです。を割り当てるには AWS EC2 インスタンスにロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルには ロールが含まれており、EC2インスタンスで実行されているプログラムが一時的な認証情報を取得できるようにします。詳細については、「 ユーザーガイド」の「 IAMロールを使用して Amazon EC2インスタンスで実行されているアプリケーションにアクセス許可を付与するIAM」を参照してください。

IAM ロールとIAMユーザーのどちらを使用するかについては、「 ユーザーガイド」の「 (ユーザーではなく) IAMロールを作成する場合IAM」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスをコントロールする AWS ポリシーを作成して にアタッチする AWS ID またはリソース。ポリシーは のオブジェクトです。 AWS アイデンティティまたはリソースに関連付けられている場合、そのアクセス許可を定義します。 AWS は、プリンシパル (ユーザー、ルートユーザー、ま

たはロールセッション) がリクエストを行うときに、これらのポリシーを評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは に保存されます。 AWS JSON ドキュメントとして。JSON ポリシードキュメントの構造と内容の詳細については、「 ユーザーガイド」のJSON「ポリシーの概要IAM」を参照してください。

管理者は を使用できます AWS JSON ポリシーは、誰が何にアクセスできるかを指定します。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するために、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行に使用するメソッドに関係なく、アクションのアクセス許可を定義します。例えば、`iam:GetRole`アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 からロール情報を取得できます。 AWS Management Console、 AWS CLI、または AWS API.

アイデンティティベースのポリシー

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできるJSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「 ユーザーガイド」のIAM「ポリシーの作成IAM」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです。 AWS アカウント。 管理ポリシーには以下が含まれます。 AWS 管理ポリシーとカスタマー管理ポリシー。管理ポリシーとインラインポリシーのどちらかを選択する方法については、IAM ユーザーガイドの「管理ポリシーとインラインポリシーの選択」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースのポリシーの例としては、IAMロールの信頼ポリシー や Amazon S3 バケットポリシー などがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があり

ます。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS のサービス.

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。は使用できません AWS リソースベースのポリシーIAMの からの マネージドポリシー。

アクセスコントロールリスト (ACLs)

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLs はリソースベースのポリシーに似ていますが、JSONポリシードキュメント形式を使用しません。

Amazon S3、 AWS WAF、、および Amazon VPCは、 をサポートする のサービスの例ですACLs。の詳細についてはACLs、Amazon Simple Storage Service デベロッパーガイドの「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 – アクセス許可の境界は、アイデンティティベースのポリシーがIAMエンティティ (IAMユーザーまたはロール) に付与できるアクセス許可の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「 IAMユーザーガイド」の「 IAMエンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) — SCPsは、 の組織または組織単位 (OU) の最大アクセス許可を指定するJSONポリシーです。 AWS Organizations. AWS Organizations は、複数の をグループ化して一元管理するためのサービスです。 AWS アカウント お客様のビジネスが所有する 。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) をアカウントの一部またはすべてに適用できます。は、各 を含むメンバーアカウントのエンティティのアクセス許可SCPを制限します。 AWS アカウントのルートユーザー。 Organizations と の詳細についてはSCPs、「」の「サービスコントロールポリシー」を参照してください。 AWS Organizations ユーザーガイド 。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果として

セッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「 ユーザーガイド」の「セッションポリシーIAM」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。方法を学ぶには AWS は、複数のポリシータイプが関与する場合にリクエストを許可するかどうかを決定します。「 ユーザーガイド」の「ポリシー評価ロジックIAM」を参照してください。

Verified Access と の連携方法 IAM

IAM を使用して Verified Access へのアクセスを管理する前に、Verified Access で使用できるIAM機能を確認してください。

IAM 機能	Verified Access サポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	Yes
ポリシー条件キー	あり
ACLs	なし
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	あり
プリンシパル権限	あり
サービスロール	いいえ

IAM 機能	Verified Access サポート
サービスリンクロール	あり

Verified Access とその他の の概要を把握するには AWS サービスはほとんどのIAM機能で動作します。「」を参照してください。 AWS ユーザーガイドIAMの「 と連携する IAM のサービス」。

Verified Access のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできるJSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「 ユーザーガイド」のIAM「ポリシーの作成IAM」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否されたアクションとリソース、およびアクションが許可または拒否される条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「 ユーザーガイド」の「 IAMJSONポリシー要素のリファレンスIAM」を参照してください。

Verified Access のアイデンティティベースポリシーの例

Verified Access アイデンティティベースポリシーの例を表示するには、「Verified Access のアイデンティティベースポリシーの例」を参照してください。

Verified Access 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースのポリシーの例としては、IAMロールの信頼ポリシー や Amazon S3 バケットポリシー などがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS のサービス.

クロスアカウントアクセスを有効にするには、リソースベースのポリシーのプリンシパルとして、アカウント全体または別のアカウントのIAMエンティティを指定できます。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる場合 AWS アカウント、信頼されたアカウントのIAM管理者は、リソースにアクセスするためのアクセス許可をプリンシパルエンティティ (ユーザーまたはロール) に付与する必要もあります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、「 ユーザーガイド」の「 でのクロスアカウントリソースアクセスIAMIAM」を参照してください。

Verified Access のポリシーアクション

ポリシーアクションのサポート: あり

管理者は を使用できます AWS JSON ポリシーは、誰が何にアクセスできるかを指定します。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action`要素は、ポリシーでアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションは通常、関連付けられている と同じ名前です。 AWS API オペレーション。一致するAPIオペレーションがないアクセス許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Verified Access アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon で定義されるアクションEC2」を参照してください。

Verified Access のポリシーアクションは、アクションの前に以下のプレフィックスを使用します。

```
ec2
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
```

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

Verified Access アイデンティティベースポリシーの例を表示するには、「Verified Access のアイデンティティベースポリシーの例」を参照してください。

Verified Access のポリシーリソース

ポリシーリソースのサポート: あり

管理者は を使用できます AWS JSON ポリシーは、誰が何にアクセスできるかを指定します。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Policy ResourceJSON要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

Verified Access リソースのタイプとその のリストを確認するにはARNs、「サービス認証リファレンス」の「Amazon で定義されるリソースEC2」を参照してください。各リソースARNの を指定できるアクションについては、「Amazon で定義されるアクションEC2」を参照してください。

Verified Access アイデンティティベースポリシーの例を表示するには、「Verified Access のアイデンティティベースポリシーの例」を参照してください。

Verified Access の条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は を使用できます AWS JSON ポリシーは、誰が何にアクセスできるかを指定します。つまり、どのプリンシパルが、どのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

ステートメントで複数のCondition要素を指定するか、単一のCondition要素で複数のキーを指定する場合は、 AWS は論理ANDオペレーションを使用してそれらを評価します。1 つの条件キーに複数の値を指定する場合は、 AWS は論理ORオペレーションを使用して条件を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば、リソースにIAMユーザー名でタグ付けされている場合にのみ、リソースへのアクセス許可をIAMユーザーに付与できます。詳細については、「 ユーザーガイド」のIAM「ポリシー要素: 変数とタグIAM」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべてを表示するには AWS グローバル条件キー、「」を参照してください。 AWSIAM ユーザーガイドの グローバル条件コンテキストキー。

Verified Access の条件キーのリストを確認するには、「サービス認証リファレンス」の「Amazon の条件キーEC2」を参照してください。条件キーを使用できるアクションとリソースについては、「Amazon で定義されるアクションEC2」を参照してください。

Verified Access アイデンティティベースポリシーの例を表示するには、「Verified Access のアイデンティティベースポリシーの例」を参照してください。

ACLs Verified Access の

をサポートACLs： いいえ

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLs はリソースベースのポリシーに似ていますが、JSONポリシードキュメント形式を使用しません。

ABAC Verified Access を使用する

サポート ABAC (ポリシー内のタグ): 部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。In AWSでは、これらの属性はタグ と呼ばれます。IAM エンティティ (ユーザーまたはロール) と多くの にタグをアタッチできます。 AWS リソースの使用料金を見積もることができます。エンティティとリソースのタグ付けは、 の最初のステップですABAC。次に、プリンシパルのタグが、アクセスしようとしているリソースのタグと一致する場合に、オペレーションを許可するABACポリシーを設計します。

ABAC は、急速に成長している環境や、ポリシー管理が煩雑になる状況に役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

の詳細についてはABAC、「 ユーザーガイド」の「 とはABACIAM」を参照してください。のセットアップ手順を含むチュートリアルを表示するにはABAC、「 ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用するIAM」を参照してください。

Verified Access での一時的認証情報の使用

一時的な認証情報のサポート: あり

ある程度 AWS のサービス 一時的な認証情報を使用してサインインすると、 は機能しません。以下を含む追加情報 AWS のサービス 一時的な認証情報の使用については、「」を参照してください。AWS のサービス ユーザーガイドの IAMで動作する IAM 。

にサインインする場合、一時的な認証情報を使用している AWS Management Console ユーザー名とパスワード以外の方法を使用する。例えば、 にアクセスする場合 AWS 会社のシングルサインオン (SSO) リンクを使用すると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えの詳細については、「 IAMユーザーガイド」の「ロールへの切り替え (コンソール) 」を参照してください。

を使用して、一時的な認証情報を手動で作成できます。 AWS CLI または AWS API。その後、これらの一時的な認証情報を使用して にアクセスできます。 AWS. AWS では、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「」の「一時的なセキュリティ認証情報IAM」を参照してください。

Verified Access のクロスサービスプリンシパル許可

転送アクセスセッションをサポート (FAS): はい

IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を使用します。 AWS のサービス、 リクエストとの組み合わせ AWS のサービス ダウンストリームサービスにリクエストを行う 。FAS リクエストは、サービスが他の とのやり取りを必要とするリクエストを受け取った場合にのみ行われます。 AWS のサービス または完了するリソース。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Verified Access のサービスロール

サービスロールのサポート: なし

サービスロールは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、 内からサービスロールを作成、変更、削除できますIAM。詳細については、「 にアクセス許可を委任するロールの作成」を参照してください。 AWS のサービス「」(IAM ユーザーガイド) を参照してください。

Verified Access 用のサービスにリンクされたロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です。 AWS のサービス。 このサービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールが に表示されます。 AWS アカウント と は サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

Verified Access のサービスにリンクされたロールの作成または管理の詳細については、「Verified Access のサービスにリンクされたロールを使用する」を参照してください。

Verified Access のアイデンティティベースポリシーの例

デフォルトでは、ユーザーおよびロールには、Verified Access リソースを作成または変更するアクセス許可はありません。また、 を使用してタスクを実行することはできません。 AWS Management

Console, AWS Command Line Interface (AWS CLI) 、または AWS API。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するために、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

これらのポリシードキュメント例を使用してIAMアイデンティティベースのJSONポリシーを作成する方法については、「 ユーザーガイド」のIAM「ポリシーの作成IAM」を参照してください。

ARNs 各リソースタイプの の形式など、Verified Access で定義されるアクションとリソースタイプの詳細については、「サービス認証リファレンス」の「Amazon のアクション、リソース、および条件キーEC2」を参照してください。

トピック

- ポリシーのベストプラクティス
- Verified Access インスタンスを作成するためのポリシー
- 自分の権限の表示をユーザーに許可する

ポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウントで誰かが Verified Access リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、 のコストが発生する可能性があります。 AWS アカウント。 アイデンティティベースのポリシーを作成または編集するときは、以下のガイドラインと推奨事項に従ってください。

- の使用を開始する AWS 管理ポリシーと最小特権のアクセス許可への移行 – ユーザーとワークロードへのアクセス許可の付与を開始するには、 を使用します。 AWS 多くの一般的なユースケースにアクセス許可を付与する マネージドポリシー。これらは で利用できます。 AWS アカウント。 を定義してアクセス許可をさらに減らすことをお勧めします。 AWS ユースケースに固有の カスタマー管理ポリシー。詳細については、「」を参照してくださいAWS マネージドポリシーまたは AWS ユーザーガイドの ジョブ機能の IAM マネージドポリシー。
- 最小特権のアクセス許可を適用する – IAMポリシーでアクセス許可を設定する場合は、タスクの実行に必要なアクセス許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用してアクセス許可を適用する方法の詳細については、「 ユーザーガイド」の「 のポリシーとアクセス許可IAMIAM」を参照してください。
- IAM ポリシーの条件を使用してアクセスをさらに制限する – ポリシーに条件を追加して、アクションとリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリ

クエストを を使用して送信する必要があることを指定できますSSL。特定の を通じてサービスアクションが使用されている場合、条件を使用してサービスアクションへのアクセスを許可することもできます。 AWS のサービスまたは AWS CloudFormation。 詳細については、「 ユーザーガイド」のIAMJSON「ポリシー要素: 条件IAM」を参照してください。

- IAM Access Analyzer を使用してIAMポリシーを検証し、安全で機能的なアクセス許可を確保する – IAM Access Analyzer は、ポリシーがポリシー言語 (JSON) とIAMベストプラクティスに準拠するように、新規および既存のIAMポリシーを検証します。IAM Access Analyzer には、安全で機能的なポリシーの作成に役立つ 100 を超えるポリシーチェックと実用的な推奨事項が用意されています。詳細については、「 ユーザーガイド」のIAM「Access Analyzer ポリシーの検証IAM」を参照してください。
- 多要素認証を要求する (MFA) – でIAMユーザーまたはルートユーザーを必要とするシナリオがある場合 AWS アカウントのセキュリティを強化するMFAには、 をオンにします。API オペレーションが呼び出されるMFAタイミングを要求するには、ポリシーにMFA条件を追加します。詳細については、「 IAMユーザーガイド」のMFA「 で保護されたAPIアクセスの設定」を参照してください。

のベストプラクティスの詳細についてはIAM、「 ユーザーガイド」の「 のセキュリティのベストプラクティスIAMIAM」を参照してください。

Verified Access インスタンスを作成するためのポリシー

Verified Access インスタンスを作成するには、IAMプリンシパルがIAMポリシーにこの追加のステートメントを追加する必要があります。

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

> **Note**
>
> `verified-access:AllowVerifiedAccess` はアクションのみの仮想 です API。リソース、タグ、または条件キーベースの認証はサポートされていません。`ec2:CreateVerifiedAccessInstance` API アクションでリソース、タグ、または条件キーベースの認証を使用します。

Verified Access インスタンスを作成するためのポリシーの例。この例では、 123456789012は です。 AWS アカウント番号 および us-east-1 は です AWS リージョン。

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

自分の権限の表示をユーザーに許可する

この例では、IAMユーザーがユーザー ID にアタッチされているインラインポリシーと管理ポリシーを表示できるようにするポリシーを作成する方法を示します。このポリシーには、コンソールで、または を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。 AWS CLI または AWS API.

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

```
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Verified Access アイデンティティとアクセスに関するトラブルシューティング

以下の情報は、Verified Access および の使用時に発生する可能性がある一般的な問題の診断と修正に役立ちますIAM。

問題

- Verified Access でアクションを実行する権限がない
- iam を実行する権限がありません。PassRole
- 自分の 以外のユーザーに許可したい AWS アカウント Verified Access リソースにアクセスする

Verified Access でアクションを実行する権限がない

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次の例のエラーは、mateojacksonIAMユーザーが コンソールを使用して架空の*my-example-widget*リソースの詳細を表示しようとしているが、架空のec2:*GetWidget*アクセス許可がない場合に発生します。

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

この場合、ec2:*GetWidget* アクションを使用して *my-example-widget*リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、 にお問い合わせください。 AWS 管理者。サインイン認証情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Verified Access にロールを渡すことができるようにする必要があります。

ある程度 AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次の例のエラーは、 というIAMユーザーがコンソールを使用して Verified Access marymajor でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

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

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 にお問い合わせください。 AWS 管理者。サインイン認証情報を提供した担当者が管理者です。

自分の 以外のユーザーに許可したい AWS アカウント Verified Access リソースにアクセスする

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACLs) をサポートするサービスでは、これらのポリシーを使用して、ユーザーにリソースへのアクセスを許可できます。

詳細については、以下を参照してください。

- Verified Access でこれらの機能がサポートされるかどうかを確認するには、「Verified Access との連携方法 IAM」を参照してください。

- 全体で リソースへのアクセスを提供する方法を学ぶには AWS アカウント 所有している 。「別のIAMユーザーへのアクセスを提供する」を参照してください。 AWS アカウント ユーザーガイドで所有している IAM 。
- リソースへのアクセスをサードパーティーに提供する方法を学ぶには AWS アカウント、「 へのアクセスの提供」を参照してください。 AWS アカウント ユーザーガイドの「第三者が所有していますIAM」。
- ID フェデレーションを通じてアクセスを提供する方法については、 IAMユーザーガイドの「外部認証されたユーザーへのアクセスの提供 (ID フェデレーション)」を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、「 ユーザーガイド」の「 でのクロスアカウントリソースアクセスIAMIAM」を参照してください。

Verified Access のサービスにリンクされたロールを使用する

AWS Verified Access は AWS Identity and Access Management 、 (IAM) サービスにリンクされたロール を使用します。サービスにリンクされたロールは、Verified Access に直接リンクされた一意のタイプのIAMロールです。サービスにリンクされたロールは、Verified Access によって事前定義されており、 AWS のサービス ユーザーに代わってサービスから他の を呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用することで、必要なアクセス権限を手動で追加する必要がなくなるため、Verified Access の設定が簡単になります。Verified Access は、サービスにリンクされたロールのアクセス許可を定義します。特に定義されている場合を除き、Verified Access のみがそのロールを引き受けることができます。定義されたアクセス許可には、信頼ポリシーとアクセス許可ポリシーが含まれており、このアクセス許可ポリシーを他のIAMエンティティにアタッチすることはできません。

サービスにリンクされたロールをサポートする他のサービスの詳細については、「 AWS と連携するのサービスIAM」を参照し、「サービスにリンクされたロール」列で「はい」があるサービスを探します。 サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[Yes] (はい) リンクを選択します。

Verified Access のためのサービスにリンクされたロールの許可

Verified Access は、 という名前のサービスにリンクされたロールAWSServiceRoleForVPCVerifiedAccessを使用して、サービスを使用するために必要なリソースをアカウントにプロビジョニングします。

AWSServiceRoleForVPCVerifiedAccess サービスにリンクされたロールは、次のサービスを信頼してロールを引き受けます。

- `verified-access.amazonaws.com`

という名前のロールアクセス許可ポリシーAWSVPCVerifiedAccessServiceRolePolicyにより、Verified Access は指定されたリソースに対して次のアクションを実行できます。

- アクション `ec2:CreateNetworkInterface` すべてのサブネット、セキュリティグループ、およびタグ `VerifiedAccessManaged=true` が付いたすべてのネットワークインターフェイスで
- アクション `ec2:CreateTags` 作成時のすべてのネットワークインターフェースで
- アクション `ec2:DeleteNetworkInterface` タグ `VerifiedAccessManaged=true` が付いたすべてのネットワークインターフェースで
- アクション `ec2:ModifyNetworkInterfaceAttribute` すべてのセキュリティグループ、およびタグ `VerifiedAccessManaged=true` が付いたすべてのネットワークインターフェースで

このポリシーのアクセス許可は、「」で確認することも AWS Management Console AWSVPCVerifiedAccessServiceRolePolicy、「 マネージドAWSVPCVerifiedAccessServiceRolePolicyポリシーリファレンスガイド」で確認することもできます。 AWS

IAM エンティティ (ユーザー、グループ、ロールなど) がサービスにリンクされたロールを作成、編集、または削除できるようにするには、アクセス許可を設定する必要があります。詳細については、「 ユーザーガイド」の「サービスにリンクされたロールのアクセス許可IAM」を参照してください。

Verified Access のサービスにリンクされたロールを作成する

サービスリンクロールを手動で作成する必要はありません。、 AWS Management Console、 AWS CLIまたは CreateVerifiedAccessEndpointで を呼び出すと AWS API、Verified Access によってサービスにリンクされたロールが作成されます。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。CreateVerifiedAccessEndpoint を再度呼び出すと、Verified Access によってサービスにリンクされたロールが再度作成されます。

Verified Access のサービスにリンクされたロールを編集する

Verified Access では、AWSServiceRoleForVPCVerifiedAccessサービスにリンクされたロールを編集することはできません。サービスリンクロールを作成した後は、多くのエンティティによってロールが参照される可能性があるため、ロール名を変更することはできません。ただし、 を使用してロールの説明を編集することはできますIAM。詳細については、「 IAMユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

Verified Access のサービスにリンクされたロールを削除する

AWSServiceRoleForVPCVerifiedAccess ロールを手動で削除する必要はありません。 AWS Management Console、、 AWS CLIまたは AWS DeleteVerifiedAccessEndpointで を呼び出すと API、Verified Access はリソースをクリーンアップし、サービスにリンクされたロールを削除します。

を使用してサービスにリンクされたロールを手動で削除するには IAM

IAM コンソール、、または AWS API を使用して AWS CLI、AWSServiceRoleForVPCVerifiedAccessサービスにリンクされたロールを削除します。詳細については、「 ユーザーガイド」の「サービスにリンクされたロールの削除IAM」を参照してください。

Verified Access サービスにリンクされたロールをサポートするリージョン

Verified Access は、サービス AWS リージョン が利用可能なすべての で、サービスにリンクされたロールの使用をサポートします。詳細については、「AWS リージョンとエンドポイント」を参照してください。

AWS Verified Access の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシーを絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS 管理ポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID

(ユーザー、グループ、ロール) が更新されます。 AWS のサービス は、新しい AWS が起動されるか、既存のサービスで新しいAPIオペレーションが使用可能になると、 AWS 管理ポリシーを更新する可能性が最も高くなります。

詳細については、「 ユーザーガイド」の「 AWS 管理ポリシーIAM」を参照してください。

AWS マネージドポリシー: AWSVPCVerifiedAccessServiceRolePolicy

このポリシーは、ユーザーに代わって Verified Access がアクションを実行することを許可する、サービスにリンクされたロールに添付されます。詳細については、「サービスにリンクされたロールの使用」を参照してください。このポリシーのアクセス許可を表示するには、AWSVPCVerifiedAccessServiceRolePolicy「」の「」を参照するか AWS Management Console、「 管理AWSVPCVerifiedAccessServiceRolePolicyポリシーリファレンスガイド」の「」を参照してください。 AWS

AWS マネージドポリシーに対する Verified Access の更新

Verified Access の AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について表示します。このページの変更に関する自動通知については、Verified Access ドキュメントの履歴ページのRSSフィードにサブスクライブしてください。

変更	説明	日付
AWSVPCVerifiedAccessServiceRolePolicy - ポリシーが更新されました	Verified Access は、「sid」フィールドのすべてのアクションの説明を含めるようにマネージドポリシーを更新しました。	2023 年 11 月 17 日
AWSVPCVerifiedAccessServiceRolePolicy - ポリシーが更新されました	Verified Access は、セキュリティグループリソースを アクセスec2:CreateNetworkInterface 許可に追加するために マネージドポリシーを更新しました。	2023 年 5 月 31 日
AWSVPCVerifiedAccessServiceRolePolicy - 新しいポリシー	Verified Access に、サービスの使用に必要なリソースをアカウントにプロビジョニング	2022 年 11 月 29 日

変更	説明	日付
	できるようにする新しいポリシーが追加されました。	
Verified Access は変更の追跡を開始	Verified Access が AWS マネージドポリシーの変更の追跡を開始しました。	2022 年 11 月 29 日

Verified Access のコンプライアンス検証

AWS Verified Access は、連邦情報処理標準 (FIPS) コンプライアンスをサポートするように設定できます。Verified Access のFIPSコンプライアンスの設定に関する詳細と設定については、「」を参照してくださいFIPS Verified Access の コンプライアンス。

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス による対象範囲内のコンプライアンスプログラムを参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「 でのレポートのダウンロード AWS Artifact」の」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。 は、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- アマゾン ウェブ サービスHIPAAのセキュリティとコンプライアンスのためのアーキテクチャ – このホワイトペーパーでは、企業が AWS を使用して HIPAA対象アプリケーションを作成する方法について説明します。

> **Note**
>
> すべての AWS のサービス がHIPAA対象となるわけではありません。詳細については、HIPAA「対象サービスリファレンス」を参照してください。

- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド — コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス 、複数のフレームワーク (米国国立標準技術研究所 (NIST) 、Payment Card Industry Security Standards Council ()、PCI国際標準化機構 (ISO) など) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「 デベロッパーガイド」の「 ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、Security Hub のコントロールリファレンスを参照してください。
- Amazon GuardDuty – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。 GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことでDSS、 PCI などのさまざまなコンプライアンス要件への対応に役立ちます。
- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

Verified Access における耐障害性

AWS グローバルインフラストラクチャは、 AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。 は、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーン AWS リージョン を提供します。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

AWS リージョン およびアベイラビリティーゾーンの詳細については、AWS 「 グローバルインフラストラクチャ」を参照してください。

Verified Access には、 AWS グローバルインフラストラクチャに加えて、高可用性のニーズをサポートするために以下の機能が用意されています。

高可用性対応の複数のサブネット

ロードバランサータイプの Verified Access エンドポイントを作成する際には、エンドポイントに複数のサブネットを関連付けることができます。エンドポイントに関連付ける各サブネットは、異なるアベイラビリティーゾーンに属している必要があります。複数のサブネットを関連付けることで、複数のアベイラビリティーゾーンを使用して高い可用性を確保できます。

モニタリング AWS Verified Access

モニタリングは、 の信頼性、可用性、パフォーマンスを維持する上で重要な部分です AWS Verified Access。 は、Verified Access をモニタリングし、問題が発生したときに報告し、必要に応じて自動アクションを実行するための以下のモニタリングツール AWS を提供します。

- アクセスログ — アプリケーションへのアクセス要求に関する詳細情報を取得します。詳細については、「the section called “Verified Access ログ”」を参照してください。
- AWS CloudTrail – によって、または に代わって行われたAPI呼び出しおよび関連イベントをキャプチャ AWS アカウント し、指定した Amazon S3 バケットにログファイルを配信します。を呼び出したユーザーとアカウント AWS、呼び出し元のソース IP アドレス、呼び出しが発生した日時を特定できます。詳細については、「the section called “CloudTrail ログ”」を参照してください。

Verified Access ログ

が各アクセスリクエスト AWS Verified Access を評価すると、すべてのアクセス試行がログに記録されます。これにより、アプリケーションへのアクセスを一元的に可視化でき、セキュリティインシデントや監査リクエストに迅速に対応できます。Verified Access は、オープンサイバーセキュリティスキーマフレームワーク (OCSF) のログ記録形式をサポートしています。

ログ記録を有効にするときは、ログの送信先を設定する必要があります。ログ記録の送信先を設定するために使用されるIAMプリンシパルには、ログ記録が正しく機能するための特定のアクセス許可が必要です。各ログ記録先に必要なIAMアクセス許可は、 Verified Access ログ記録のアクセス許可セクションで確認できます。Verified Access は、以下のアクセスログのパブリッシュ先をサポートします。

- Amazon CloudWatch Logs ロググループ
- Amazon S3 バケット
- Amazon Data Firehose 配信ストリーム

内容

- Verified Access 信頼コンテキストを有効または無効にする
- OCSF Verified Access のバージョン 0.1 ログの例
- OCSF Verified Access のバージョン 1.0.0-rc.2 ログの例

Verified Access のログ記録バージョン

デフォルトでは、Verified Access ログ記録システムは Open Cybersecurity Schema Framework (OCSF) バージョン 0.1 を使用します。バージョン 0.1 を使用したサンプルログは、OCSF Verified Access のバージョン 0.1 ログの例 セクションで確認できます。

最新のログ記録バージョンは、OCSFバージョン 1.0.0-rc.2 と互換性があります。スキーマの に関する具体的な詳細については、OCSF「スキーマ」を参照してください。バージョン 1.0.0-rc.2 を使用したサンプルログは、OCSF Verified Access のバージョン 1.0.0-rc.2 ログの例 セクションで確認できます。

使用中のログ記録バージョンをアップグレードする場合は、次の手順を使用します。

コンソールを使用してロギングバージョンをアップグレードするには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. 適切な Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. 「ログバージョンの更新」ドロップダウンリストから ocsf-1.0.0-rc.2 を選択します。
6. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用してログ記録バージョンをアップグレードするには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

Verified Access ログ記録のアクセス許可

ログ記録の送信先を設定するために使用されるIAMプリンシパルには、ログ記録が正しく機能するための特定のアクセス許可が必要です。以下のセクションでは、各ログ記録先に必要なアクセス許可を示します。

CloudWatch ログへの配信の場合：

- Verified Access インスタンスの ec2:ModifyVerifiedAccessInstanceLoggingConfiguration
- すべてのリソースの logs:CreateLogDelivery、logs:DeleteLogDelivery、logs:GetLogDelivery、logs:ListLo および logs:UpdateLogDelivery
- 送信先ロググループの logs:DescribeLogGroups、logs:DescribeResourcePolicies および logs:PutResourcePolicy

Amazon S3 への配信：

- Verified Access インスタンスの ec2:ModifyVerifiedAccessInstanceLoggingConfiguration
- すべてのリソースの logs:CreateLogDelivery、logs:DeleteLogDelivery、logs:GetLogDelivery、logs:ListLo および logs:UpdateLogDelivery
- 送信先バケットの s3:GetBucketPolicy および s3:PutBucketPolicy

Firehose への配信の場合：

- Verified Access インスタンスの ec2:ModifyVerifiedAccessInstanceLoggingConfiguration
- すべてのリソースの firehose:TagDeliveryStream
- すべてのリソースの iam:CreateServiceLinkedRole
- すべてのリソースの logs:CreateLogDelivery、logs:DeleteLogDelivery、logs:GetLogDelivery、logs:ListLo および logs:UpdateLogDelivery

Verified Access ログを有効または無効にする

このセクションの手順を使用して、ログ記録を有効または無効にできます。ログ記録を有効にするときは、ログの送信先を設定する必要があります。ログ記録の送信先を設定するために使用されるIAMプリンシパルには、ログ記録が正しく機能するための特定のアクセス許可が必要です。各ログ記録

先に必要なIAMアクセス許可は、 Verified Access ログ記録のアクセス許可セクションで確認できます。

内容

アクセスログの有効化

Verified Access ログを有効にするには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. (オプション) 信頼プロバイダーから送信されるトラストデータをログに含めるには、次の操作を行います。

   a. 「ログバージョンの更新」ドロップダウンリストから ocsf-1.0.0-rc.2 を選択します。

   b. [トラストコンテキストを含める] を選択します。
6. 次のいずれかを行います。
   - Amazon CloudWatch Logs への配信を有効にします。送信先ロググループを選択します。
   - [Amazon S3 に配信] をオンにします。送信先バケットの名前、所有者、プレフィックスを入力します。
   - 「Firehose への配信」をオンにします。送信先の配信ストリームを選択します。
7. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログを有効にするには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

アクセスログの無効化

Verified Access インスタンスのアクセスログは、いつでも無効化できます。アクセスログを無効にした後は、削除するまでログデータはログ送信先に残ります。

Verified Access ログを無効にするには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. ログ配信をオフにします。
6. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログを無効にするには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

Verified Access 信頼コンテキストを有効または無効にする

信頼プロバイダーから送信された信頼コンテキストは、オプションで Verified Access ログに含めることができます。これは、アプリケーションへのアクセスを許可または拒否するポリシーを定義する場合に便利です。有効にすると、信頼コンテキストは `data`フィールドの下のログに表示されます。信頼コンテキストが無効になっている場合、 `data`フィールドは に設定されます`null`。ログに信頼コンテキストを含めるように Verified Access を設定するには、次の手順を実行します。

> **Note**
>
> Verified Access ログにトラストコンテキストを含めるには、最新の ロギングバージョン `ocsf-1.0.0-rc.2` にアップグレードする必要があります。次の手順では、ログ記録が既に有効になっていることを前提としています。そうでない場合、手順の詳細については アクセスログの有効化 を参照してください。

内容

- トラストコンテキストを有効にする
- トラストコンテキストを無効にする

トラストコンテキストを有効にする

コンソールを使用して Verified Access ログにトラストコンテキストを含めるには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. 適切な Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. 「ログバージョンの更新」ドロップダウンリストから [ocsf-1.0.0-rc.2] を選択します。
6. [トラストコンテキストを含める] をオンにします。
7. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログに信頼コンテキストを含めるには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

トラストコンテキストを無効にする

ログに信頼コンテキストを含めなくなった場合は、次の手順を実行して削除できます。

コンソールを使用して Verified Access ログからトラストコンテキストを削除するには

1. で Amazon VPCコンソールを開きますhttps://console.aws.amazon.com/vpc/。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. 適切な Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. [トラストコンテキストを含める] をオフにします。
6. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログから信頼コンテキストを削除するには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

OCSF Verified Access のバージョン 0.1 ログの例

デフォルトのログ記録OCSFバージョン 0.1 を使用したサンプルログを次に示します。

例

- で付与されたアクセス OIDC
- OIDC および で付与されるアクセス JAMF
- OIDC および で付与されたアクセス CrowdStrike
- Cookie の欠落によるアクセスの拒否
- ポリシーによるアクセス拒否
- 不明なログエントリ

で付与されたアクセス OIDC

このログエントリの例では、Verified Access はOIDCユーザー信頼プロバイダーを使用してエンドポイントへのアクセスを許可します。


```
{
    "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
}
```

OIDC および で付与されるアクセス JAMF

この口グエントリの例では、Verified Access は OIDCと JAMF デバイス信頼プロバイダーの両方を持つエンドポイントへのアクセスを許可します。

```
{
    "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
}
```


OIDC および で付与されたアクセス CrowdStrike

このログエントリの例では、Verified Access は OIDCと CrowdStrike デバイス信頼プロバイダーの両方を持つエンドポイントへのアクセスを許可します。


```
{
    "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
}
```

Cookie の欠落によるアクセスの拒否

このログエントリの例では、認証 Cookie の欠落により Verified Access がアクセスを拒否します。

```
{
    "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
}
```

ポリシーによるアクセス拒否

このログエントリの例では、認証されたリクエストがアクセスポリシーで許可されていないため、Verified Access は認証されたリクエストを拒否します。

```
{
    "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
```

```
}
```

不明なログエントリ

このログエントリの例では、Verified Access では完全なログエントリを生成できないため、不明なログエントリが出力されます。これにより、すべてのリクエストがアクセスログに表示されることが保証されます。

```
{
    "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 Verified Access のバージョン 1.0.0-rc.2 ログの例

ログ記録OCSFバージョン 1.0.0-rc.2 を使用したサンプルログを次に示します。

内容

トラストコンテキストを含むアクセス許可

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
```

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

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

```
                "locale": "US",
                "nickname": "Tester",
                "email": "johndoe-user@test.com"
            },
            "http_request": {
                "x_forwarded_for": "1.1.1.1,2.2.2.2",
                "http_method": "GET",
                "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
                "port": "80",
                "hostname": "hostname.net"
            }
        }
    }
}
```

トラストコンテキストを省略したアクセス許可

```
{
    "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
}
```


を使用した Verified Access APIコールのログ記録 AWS CloudTrail

AWS Verified Access は と統合されています AWS CloudTrail、ユーザー、ロール、または によって実行されたアクションを記録するサービス AWS のサービス Verified Access. CloudTrail captures のは、Verified Access をイベントとしてAPI呼び出します。キャプチャされた呼び出しには、Verified Access コンソールからの呼び出しと Verified Access APIオペレーションへのコード呼び出しが含まれます。で収集された情報を使用して CloudTrail、Verified Access に対するリクエスト、リクエスト元の IP アドレス、リクエスト日時などの詳細を確認できます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の によって行われたかどうか AWS のサービス.

CloudTrail は でアクティブです AWS アカウント アカウントを作成し、 CloudTrail イベント履歴に自動的にアクセスできる場合。 CloudTrail イベント履歴は、過去 90 日間に記録された管理イベントの、表示可能、検索可能、ダウンロード可能、およびイミュータブルなレコードを に提供します。 AWS リージョン。 詳細については、「」の CloudTrail 「イベント履歴の使用」を参照して

ください。 AWS CloudTrail ユーザーガイド。イベント履歴を表示するための料金はかかりません CloudTrail。

のイベントを継続的に記録するには AWS アカウント 過去 90 日間、証跡または CloudTrail Lake イベントデータストアを作成します。

CloudTrail 証跡

証跡により CloudTrail 、 はログファイルを Amazon S3 バケットに配信できます。を使用して作成されたすべての証跡 AWS Management Console はマルチリージョンです。を使用して、単一リージョンまたはマルチリージョンの証跡を作成できます。 AWS CLI。 すべての でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。 AWS リージョン アカウントの 。単一リージョンの証跡を作成する場合、証跡の に記録されたイベントのみを表示できます。 AWS リージョン。 証跡の詳細については、「 の証跡の作成」を参照してください。 AWS アカウント および での組織の証跡の作成 AWS CloudTrail ユーザーガイド。

証跡を作成 CloudTrail することで、 から Amazon S3 バケットに継続的な管理イベントのコピーを 1 つ無料で配信できますが、Amazon S3 ストレージ料金が発生します。 CloudTrail 料金の詳細については、「」を参照してください。 AWS CloudTrail 料金 Amazon S3 の料金に関する詳細については、「Amazon S3 の料金」を参照してください。

CloudTrail Lake イベントデータストア

CloudTrail Lake では、イベントに対して SQLベースのクエリを実行できます。 CloudTrail Lake は、既存のイベントを行ベースのJSON形式で Apache ORC 形式に変換します。ORC は、データを迅速に取得できるように最適化された列指向ストレージ形式です。イベントはイベントデータストアに集約されます。イベントデータストアは、高度なイベントセレクタを適用することによって選択する条件に基いた、イベントのイミュータブルなコレクションです。どのイベントが存続し、クエリに使用できるかは、イベントデータストアに適用するセレクタが制御します。CloudTrail Lake の詳細については、「 の使用」を参照してください。 AWS CloudTrail の Lake AWS CloudTrail ユーザーガイド。

CloudTrail Lake イベントデータストアとクエリにはコストが発生します。イベントデータストアを作成する際に、イベントデータストアに使用する料金オプションを選択します。料金オプションによって、イベントの取り込みと保存にかかる料金、および、そのイベントデータストアのデフォルトと最長の保持期間が決まります。 CloudTrail 料金の詳細については、「」を参照してください。 AWS CloudTrail 料金

Verified Access 管理イベント

管理イベントは、 のリソースで実行される管理オペレーションに関する情報を提供します。 AWS アカウント。 これらはコントロールプレーンオペレーションとも呼ばれます。デフォルトでは、 は管理イベント CloudTrail を記録します。

Verified Access は、コントロールプランオペレーションを管理イベントとしてログに記録します。リストについては、「Amazon EC2APIリファレンス」を参照してください。

Verified Access イベントの例

次の例は、 CreateVerifiedAccessInstanceアクションを示す CloudTrail イベントを示しています。


```
{
    "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"
}
```


CloudTrail レコードの内容の詳細については、「」のCloudTrail「レコードの内容」を参照してください。 AWS CloudTrail ユーザーガイド。

のクォータ AWS Verified Access

には、 ごとに、以前 AWS アカウント は制限と呼ばれていたデフォルトのクォータがあります AWS のサービス。特に明記されていない限り、クォータは地域固有です。

AWS アカウントレベルのクォータ

には、Verified Access に関連する以下のクォータ AWS アカウント があります。

名前	デフォルト	引き上げ可能	説明
Verified Access インスタンス	5	はい	お客様が現在のリージョンで作成できる Verified Access インスタンスの最大数。
Verified Access グループ	10	はい	お客様が現在のリージョンで作成できる Verified Access グループの最大数。
Verified Access 信頼プロバイダー	15	はい	お客様が現在のリージョンで作成できる Verified Access 信頼プロバイダーの最大数。
Verified Access エンドポイント	50	はい	お客様が現在のリージョンで作成できる Verified Access エンドポイントの最大数。

HTTP ヘッダー

HTTP ヘッダーのサイズ制限は次のとおりです。

名前	デフォルト	引き上げ可能
リクエスト行	16 K	なし
単一ヘッダー	16 K	なし

名前	デフォルト	引き上げ可能
レスポンスのヘッダー全体	32 K	なし
リクエストのヘッダー全体	64 K	なし

OIDC クレームサイズ

OIDC クレームサイズの制限は次のとおりです。

名前	デフォルト	引き上げ可能
OIDC クレームサイズ	11 K	なし

「Verified Access ユーザーガイド」のドキュメント履歴

次の表は、「Verified Access」のドキュメントリリースの内容をまとめたものです。

変更	説明	日付
AWS マネージドポリシーの更新	Verified Access の AWS マネージドIAMポリシーを更新しました。	2023 年 11 月 17 日
保管時のデータ暗号化	AWS Verified Access は、デフォルトで AWS 所有KMS キーを使用して保管中のデータを暗号化します。	2023 年 9 月 28 日
FIPS コンプライアンスのサポート	FIPS コンプライアンスのために Verified Access を設定します。	2023 年 9 月 26 日
高度なログ記録	ログにトラストコンテキストを追加するログ記録機能の追加。	2023 年 6 月 19 日
AWS マネージドポリシーの更新	Verified Access の AWS マネージドIAMポリシーを更新しました。	2023 年 5 月 31 日
GA リリース	「Verified Access ユーザーガイド」の GA リリース。AWS WAF 統合を含んでいます。	2023 年 4 月 27 日
プレビューリリース	「Verified Access ユーザーガイド」のプレビューリリース	2022 年 11 月 29 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。