사용자 가이드

AWS 검증된 액세스

AWS 검증된 액세스: 사용자 가이드

Table of Contents

무엇입니까 AWS Verified Access?

를 사용하면 가상 사설망 (VPN) 을 사용하지 않고도 애플리케이션에 대한 보안 액세스를 제공할 수 있습니다. AWS Verified Access Verified Access는 각 애플리케이션 요청을 평가하여 사용자가 지정된 보안 요구 사항을 충족하는 경우에만 각 애플리케이션에 액세스할 수 있도록 합니다.

Verified·Access의 이점

- 보안 상태 개선 - 기존 보안 모델은 액세스를 한 번 평가하여 사용자에게 모든 애플리케이션에 대한 액세스 권한을 부여합니다. Verified·Access는 각 애플리케이션 액세스 요청을 실시간으로 평가합니다. 이로 인해 악의적인 공격자가 한 애플리케이션에서 다른 애플리케이션으로 이동하기가 어렵습니다.
- 보안 서비스와의 통합 — Verified Access는 두 서비스 및 타사 서비스를 포함한 ID AWS 및 장치 관리 서비스와 통합됩니다. Verified Access는 이러한 서비스의 데이터를 사용하여 보안 요구 사항 집합을 기준으로 사용자와 디바이스의 신뢰성을 확인하고 사용자가 애플리케이션에 액세스할 수 있는지 여부를 결정합니다.
- 사용자 경험 개선 — Verified Access를 사용하면 사용자가 a를 사용하여 애플리케이션에 액세스할 VPN 필요가 없습니다. 이렇게 하면 VPN 관련 문제로 인해 발생하는 지원 사례의 수를 줄이는 데 도움이 됩니다.
- 간소화된 문제 해결 및 감사 - Verified Access는 모든 액세스 시도를 기록하여 애플리케이션 액세스에 대한 중앙 집중식 가시성을 제공하여 보안 사고 및 감사 요청에 신속하게 대응할 수 있도록 지원합니다.

Verified·Access 액세스

다음 인터페이스 중 하나를 사용하여 Verified·Access로 작업할 수 있습니다.

- AWS Management Console – Verified·Access 리소스를 생성하고 관리하는 데 사용할 수 있는 웹 인터페이스를 제공합니다. 에서 Amazon VPC 콘솔에 AWS Management Console 로그인하고 엽니다 https://console.aws.amazon.com/vpc/.
- AWS Command Line Interface (AWS CLI) — 다음을 AWS 서비스비롯한 다양한 항목에 대한 명령을 제공합니다 AWS Verified Access. AWS CLI 는 윈도우, macOS, 리눅스에서 지원됩니다. AWS CLI다운로드하려면 을 참조하십시오 AWS Command Line Interface.
- AWS SDKs— 언어별로 APIs 제공하세요. 서명 계산, 요청 재시도 및 오류 처리와 같은 많은 연결 세부 정보를 AWS SDKs 처리합니다. 자세한 내용은 AWS SDKs를 참조하세요.

- 쿼리 API — 요청을 사용하여 호출하는 저수준 API 작업을 제공합니다. HTTPS 쿼리를 사용하는 API 것이 검증된 액세스에 액세스하는 가장 직접적인 방법입니다. 하지만 이를 사용하려면 애플리케이션에서 요청에 서명할 해시 생성 및 오류 처리와 같은 하위 수준의 세부 정보를 처리해야 합니다. 자세한 내용은 Amazon EC2 API 참조의 검증된 액세스 작업을 참조하십시오.

이 가이드에서는 를 사용하여 검증된 액세스 리소스를 생성, 액세스 및 관리하는 방법을 설명합니다. AWS Management Console

요금

Verified Access의 각 애플리케이션에 대해 시간당 요금이 청구되며, Verified Access에서 처리된 데이터 양에 대한 요금이 청구됩니다. 자세한 내용은 AWS Verified Access 요금을 참조하세요.

Verified·Access의 작동 방식

AWS Verified Access 사용자의 각 애플리케이션 요청을 평가하고 다음을 기준으로 액세스를 허용합니다.

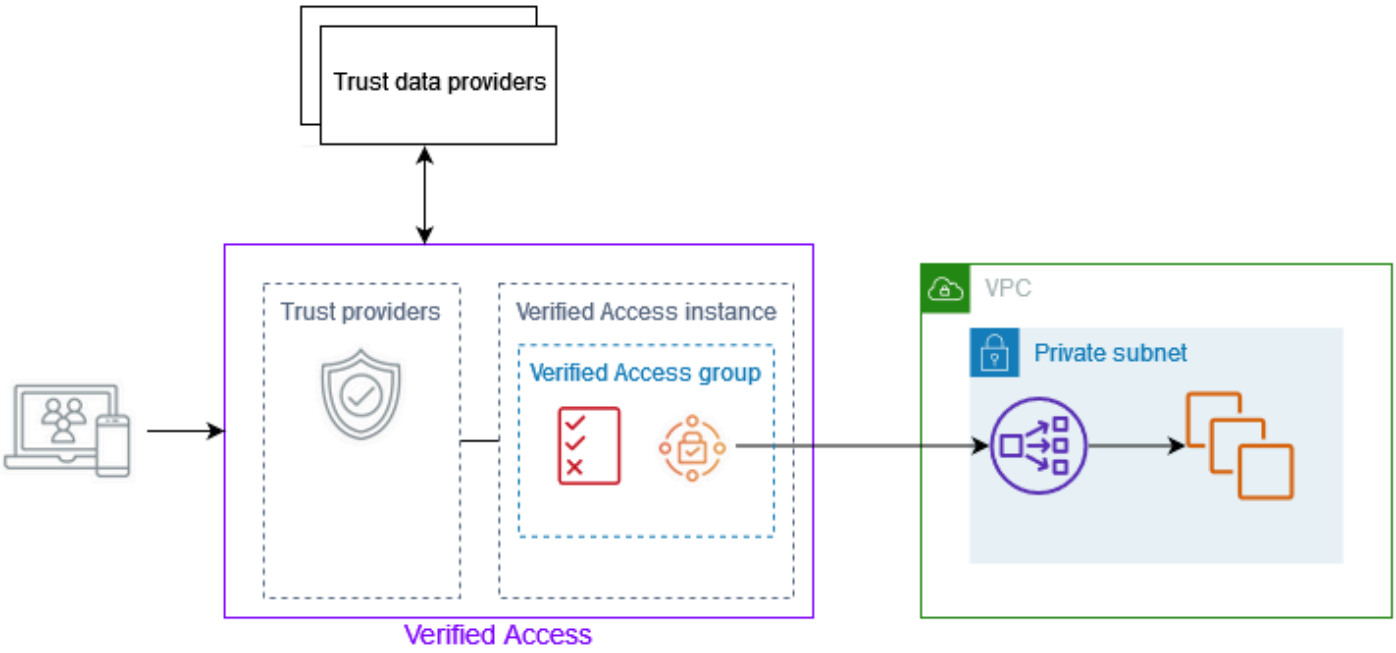
- 선택한 신뢰 제공자 (AWS 또는 제3자로부터) 가 보낸 신뢰 데이터.
- Verified·Access에서 생성한 액세스 정책.

사용자가 애플리케이션에 액세스하려고 하면 Verified·Access는 신뢰 공급자로부터 데이터를 가져와 애플리케이션에 대해 설정한 정책과 비교하여 평가합니다. Verified·Access는 사용자가 지정된 보안 요구 사항을 충족하는 경우에만 요청된 애플리케이션에 대한 액세스 권한을 부여합니다. 정책이 정의될 때까지 모든 애플리케이션 요청은 기본적으로 거부됩니다.

또한 Verified·Access는 모든 액세스 시도를 기록하므로 보안 사고 및 감사 요청에 신속하게 대응할 수 있습니다.

Verified·Access의 주요 구성 요소

다음은 Verified·Access의 중요한 개요를 설명하는 다이어그램입니다. 사용자가 애플리케이션 액세스 요청을 보냅니다. Verified·Access는 그룹에 대한 액세스 정책 및 애플리케이션별 엔드포인트 정책을 기준으로 요청을 평가합니다. 액세스가 허용되면 해당 요청이 엔드포인트를 통해 애플리케이션에 전송됩니다.



- Verified·Access 인스턴스 – 인스턴스는 애플리케이션 요청을 평가하여 보안 요구 사항이 충족되는 경우에만 액세스를 부여합니다.
- Verified·Access 엔드포인트 - 각 엔드포인트는 애플리케이션을 나타냅니다. 로드 밸런서 엔드포인트 또는 네트워크 인터페이스 엔드포인트를 생성할 수 있습니다.
- Verified·Access 그룹 – Verified·Access 엔드포인트의 모음입니다. 정책 관리를 단순화하려면 보안 요구 사항이 비슷한 애플리케이션의 엔드포인트를 그룹화하는 것이 좋습니다. 예를 들어, 모든 판매 애플리케이션의 엔드포인트를 함께 그룹화할 수 있습니다.
- 액세스 정책 - 애플리케이션에 대한 액세스를 허용할지 거부할지를 결정하는 사용자 정의 규칙 집합입니다. 사용자 ID 및 디바이스 보안 상태를 비롯한 여러 요소를 조합하여 지정할 수 있습니다. 각 Verified·Access 그룹에 대해 그룹 액세스 정책을 생성합니다. 이 정책은 그룹의 모든 엔드포인트에 상속됩니다. 선택적으로 애플리케이션별 정책을 생성하여 특정 엔드포인트에 연결할 수 있습니다.
- 신뢰 공급자 - 사용자 ID 또는 디바이스 보안 상태를 관리하는 서비스입니다. Verified Access는 두 신뢰 제공업체 AWS 및 타사 신뢰 제공자와 함께 사용할 수 있습니다. 각 Verified·Access 인스턴스에 하나 이상의 신뢰 공급자를 연결해야 합니다. 각 Verified·Access 인스턴스에 단일 ID 신뢰 공급자와 여러 디바이스 신뢰 공급자를 연결할 수 있습니다.
- 신뢰 데이터 – 신뢰 공급자가 Verified·Access에 보내는 사용자 또는 디바이스의 보안 관련 데이터입니다. 사용자 클레임또는 신뢰 컨텍스트라고도 합니다. 사용자의 이메일 주소 또는 디바이스의 운영 체제 버전을 예로 들 수 있습니다. Verified·Access는 각 애플리케이션 액세스 요청을 받을 때 액세스 정책을 기준으로 이 데이터를 평가합니다.

자습서: 검증된 액세스 시작하기

이 튜토리얼을 사용하여 시작해 보세요 AWS Verified Access. Verified·Access 리소스를 생성하고 구성하는 방법을 알아봅니다.

이 자습서에서는 Verified Access에 애플리케이션을 추가해 보겠습니다. 자습서가 끝나면 특정 사용자는 사용하지 않고도 인터넷을 통해 해당 응용 프로그램에 액세스할 수 VPN 있습니다.

> **Note**
>
> 이 자습서에서는 장치 기반 신뢰 공급자와의 통합을 설명하지 않습니다. 대신 당사는 ID 기반 신뢰 공급자와만 협력합니다.

Tasks

Verified Access 튜토리얼 사전 요구 사항

이 자습서를 완료하기 위한 사전 요구 사항은 다음과 같습니다.

- 두 가지 사용 가능 여부. AWS 계정한 계정은 대상 애플리케이션을 호스팅하고 다른 계정에는 Verified Access 리소스가 생성됩니다.

- AWS IAM Identity Center 작업 AWS 리전 중인 계정에서 활성화됩니다. 그러면 IAM Identity Center를 검증된 액세스 권한이 있는 신뢰 공급자로 사용할 수 있습니다. 자세한 내용은 사용 설명서의 IAM AWS IAM Identity Center Identity Center 활성화를 참조하십시오.
- 퍼블릭 호스팅 도메인 및 도메인의 DNS 레코드를 업데이트하는 데 필요한 권한.
- AWS 계정의 내부 로드 밸런서 뒤에서 실행되는 애플리케이션입니다. 사용할 예제 애플리케이션 도메인 이름은 `www.myapp.example.com`입니다.
- AWS Verified Access 인스턴스를 만드는 데 필요한 모든 권한을 포함하는 IAM 정책이 여기에 나와 Verified·Access 인스턴스 생성 정책 있습니다.

1단계: Verified·Access 인스턴스 생성

다음 절차에 따라 Verified·Access 인스턴스를 생성하십시오.

Verified·Access 인스턴스를 생성하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. Amazon VPC 탐색 창에서 검증된 액세스 인스턴스를 선택한 다음 검증된 액세스 인스턴스 생성을 선택합니다.
3. (선택 사항) 이름 및 설명에 Verified·Access 인스턴스의 이름과 설명을 입력합니다.
4. 신뢰 공급자의 경우 기본 옵션을 유지하십시오.
5. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
6. Verified·Access 인스턴스 생성을 선택합니다.

2단계: 검증된 액세스 신뢰 제공자 구성

신뢰 AWS IAM Identity Center 공급자로 설정할 수 있습니다.

IAMID 센터 신뢰 공급자를 만들려면

1. Amazon VPC 탐색 창에서 검증된 액세스 신뢰 공급자를 선택한 다음 검증된 액세스 신뢰 제공자 생성을 선택합니다.
2. (선택 사항) 이름 태그 및 설명에 Verified·Access 신뢰 공급자의 이름과 설명을 입력합니다.
3. 나중에 정책 참조 이름에 대한 정책 규칙 작업 시 사용할 사용자 지정 식별자를 입력합니다. 예를 들면 **`idc`**를 입력할 수 있습니다.
4. 신뢰 공급자 유형에서 사용자 신뢰 공급자를 선택합니다.

5. 사용자 신뢰 제공자 유형에서 IAM ID 센터를 선택합니다.
6. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
7. Verified·Access 신뢰 공급자 생성을 선택합니다.

3단계: 신뢰할 수 있는 공급자를 검증된 액세스 인스턴스에 연결

이제 신뢰 공급자를 구성했으니 이전에 만든 Verified Access 인스턴스에 연결할 수 있습니다. 다음 절차에 따라 Verified·Access 인스턴스에 신뢰 공급자를 연결합니다.

인스턴스에 신뢰 공급자를 연결하려면

1. Amazon VPC 탐색 창에서 검증된 액세스 인스턴스를 선택합니다.
2. 인스턴스를 선택합니다.
3. 작업, Verified·Access 신뢰 공급자 연결을 선택합니다.
4. Verified·Access 신뢰 공급자에서 신뢰 공급자를 선택합니다.
5. Verified·Access 신뢰 공급자 연결을 선택합니다.

4단계: 검증된 액세스 그룹 생성

이 단계에서는 5단계에서 엔드포인트로 사용할 그룹을 생성합니다.

Verified·Access 그룹을 생성하려면

1. Amazon VPC 탐색 창에서 검증된 액세스 그룹을 선택한 다음 검증된 액세스 그룹 생성을 선택합니다.
2. (선택 사항) 이름 태그 및 설명에 그룹의 이름과 설명을 입력합니다.
3. Verified·Access 인스턴스에서 Verified·Access 인스턴스를 선택합니다.
4. 정책 정의의 경우 이 필드를 비워 둡니다. 이 자습서의 후반에서 정책을 생성합니다.
5. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
6. Verified·Access 그룹 생성을 선택합니다.

5단계: 다음을 통해 검증된 액세스 그룹을 공유하십시오. AWS Resource Access Manager

이 단계에서는 방금 만든 그룹을 대상 애플리케이션이 실행 중인 그룹과 공유합니다. AWS 계정 Verified·Access 그룹을 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유가 없는 경우 먼저 리소스 공유를 생성해야 합니다.

에 AWS Organizations속해 있고 조직 내 공유가 활성화되어 있는 경우 조직의 소비자에게 공유된 Verified Access 그룹에 대한 액세스 권한이 자동으로 부여됩니다. 그렇지 않으면 소비자는 리소스 공유에 가입하라는 초대장을 받고 초대를 수락한 후 공유된 Verified·Access 그룹에 대한 액세스 권한이 부여됩니다.

AWS RAM 사용 설명서의 리소스 공유 생성 단계를 따릅니다. 리소스 유형 선택에서 Verified·Access 그룹을 선택한 다음 Verified·Access 그룹의 확인란을 선택합니다.

자세한 내용은 AWS RAM 사용 설명서에서 시작하기를 참조하십시오.

6단계: 검증된 액세스 엔드포인트를 생성하여 애플리케이션을 추가합니다.

다음 절차를 사용하여 검증된 액세스 엔드포인트를 생성합니다. 이 단계에서는 Elastic Load Balancing의 내부 로드 밸런서 뒤에서 실행되는 애플리케이션이 있다고 가정합니다.

Verified·Access 엔드포인트를 생성하려면

1. Amazon VPC 탐색 창에서 검증된 액세스 엔드포인트를 선택한 다음 검증된 액세스 엔드포인트 생성을 선택합니다.
2. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
3. Verified·Access 그룹에서 Verified·Access 그룹을 선택합니다.
4. 애플리케이션 세부 정보를 보려면 다음을 수행합니다.
   a. 애플리케이션 도메인의 경우 애플리케이션 DNS 이름을 입력합니다.
   b. 도메인 인증서에서 ARN 공개 TLS 인증서의 Amazon 리소스 이름 (ARN) 을 선택합니다.
5. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 첨부 유형에서 선택합니다 VPC.

b. 보안 그룹에서 엔드포인트와 연결할 보안 그룹을 선택합니다.

c. 엔드포인트 도메인 접두사에 사용자 지정 식별자를 입력합니다. 이 이름은 Verified Access에서 DNS 생성한 이름 앞에 추가됩니다. 이 예에서는 **my-ava-app**을 사용할 수 있습니다.

d. 엔드포인트 유형에서 로드 밸런서를 선택합니다.

e. 프로토콜의 경우 또는 를 선택합니다 HTTPS. HTTP 이는 로드 밸런서의 구성에 따라 달라집니다.

f. 포트에 포트 번호를 입력합니다. 이는 로드 밸런서의 구성에 따라 달라집니다.

g. 로드 밸런서에서 로드 ARN 밸런서를 선택합니다.

h. 서브넷에서 로드 밸런서와 연결된 서브넷을 선택합니다.

6. 정책 정의에는 지금은 정책을 입력하지 마십시오. 나중에 이 자습서에서 이를 다룰 것입니다.
7. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
8. Verified·Access 엔드포인트 생성을 선택합니다.

7단계: 검증된 액세스 DNS 엔드포인트에 대한 설정 구성

이 단계에서는 애플리케이션의 도메인 이름(예: www.myapp.example.com)을 Verified·Access 엔드포인트의 도메인 이름에 매핑합니다. DNS매핑을 완료하려면 공급자와 함께 표준 이름 레코드 (CNAME)를 생성하십시오. DNS CNAME레코드를 생성하면 사용자가 애플리케이션에 보내는 모든 요청이 Verified Access로 전송됩니다.

엔드포인트의 도메인 이름을 얻으려면

1. Amazon VPC 탐색 창에서 검증된 액세스 엔드포인트를 선택합니다.
2. 이전에 생성한 엔드포인트를 선택합니다.
3. 엔드포인트의 세부 정보 탭을 선택합니다.
4. 엔드포인트 도메인에서 엔드포인트 도메인을 복사합니다.

이 자습서의 엔드포인트 도메인 이름은 `my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`입니다.

DNS공급자와 함께 CNAME 레코드 생성:

레코드 이름	유형	값
www.myapp.example.com	CNAME	my-ava-app.edge-1a 2b3c4d5e6f7g.vai-1 a2b3c4d5e6f7g.prod. verified-access.us-west-2.amazonaws.com

8단계: 검증된 액세스에 추가한 애플리케이션에 대한 연결 테스트

이제 애플리케이션 연결을 테스트할 수 있습니다. 웹 브라우저에 애플리케이션의 도메인 이름을 입력합니다. Verified·Access 정책의 기본 동작은 모든 요청을 거부하는 것입니다. 누구나 액세스할 수 있는 정책을 아직 마련하지 않았으므로 모든 요청을 거부해야 합니다.

9단계: 검증된 액세스 그룹 수준 액세스 정책 구성

다음 절차에 따라 Verified·Access 그룹을 수정하고 애플리케이션에 대한 연결을 허용하는 액세스 정책을 구성합니다. 정책의 세부 사항은 IAM Identity Center에 구성된 사용자 및 그룹에 따라 달라집니다. 정책 생성에 대한 자세한 내용은 Verified·Access 정책 섹션을 참조하십시오.

Verified·Access 그룹을 수정하려면

1. Amazon VPC 탐색 창에서 검증된 액세스 그룹을 선택합니다.
2. 그룹을 선택합니다.
3. 작업, Verified·Access 그룹 정책 수정을 선택합니다.
4. 정책을 입력합니다.
5. Verified·Access 그룹 정책 수정을 선택합니다.

10단계: 검증된 액세스에 추가한 애플리케이션에 대한 연결을 다시 테스트합니다.

이제 그룹 정책이 적용되었으므로 애플리케이션에 액세스할 수 있습니다. 웹 브라우저에 애플리케이션의 도메인 이름을 입력합니다. 요청이 허용되어야 하며 애플리케이션으로 리디렉션되어야 합니다.

생성한 Verified Access 리소스를 정리하세요.

테스트를 마친 후 아래 단계에 따라 생성된 리소스를 삭제합니다.

이 자습서에서 생성한 Verified·Access 리소스를 삭제하려면

1. Amazon VPC 탐색 창에서 검증된 액세스 엔드포인트를 선택합니다. 제거할 엔드포인트를 선택합니다. 작업, Verified·Access 엔드포인트 삭제를 선택합니다.
2. 탐색 창에서 Verified·Access 그룹을 선택합니다. 제거할 그룹을 선택합니다. 작업, Verified·Access 그룹 삭제를 선택합니다. 참고 – 엔드포인트 삭제 프로세스가 완료될 때까지 몇 분 정도 기다려야 할 수 있습니다.
3. Amazon VPC 탐색 창에서 검증된 액세스 인스턴스를 선택합니다. 이 자습서용으로 생성한 인스턴스를 선택합니다. 작업, Verified·Access 신뢰 공급자 분리를 선택합니다. 드롭다운 목록에서 신뢰 공급자를 선택하고 Verified·Access 신뢰 공급자 분리를 선택합니다.
4. Amazon VPC 탐색 창에서 검증된 액세스 신뢰 공급자를 선택합니다. 이 자습서용으로 생성한 신뢰 공급자를 선택합니다. 작업, Verified·Access 신뢰 공급자 삭제를 선택합니다.
5. Amazon VPC 탐색 창에서 검증된 액세스 인스턴스를 선택합니다. 이 자습서용으로 생성한 인스턴스를 선택합니다. 작업, Verified·Access 인스턴스 삭제를 선택합니다.

Verified·Access 인스턴스

AWS Verified Access 인스턴스는 신뢰 제공자와 Verified Access 그룹을 구성하는 데 도움이 되는 AWS 리소스입니다. 인스턴스는 애플리케이션 요청을 평가하여 보안 요구 사항이 충족되는 경우에만 액세스 권한을 부여합니다.

주제

Verified Access 인스턴스 생성 및 관리

Verified Access 인스턴스를 사용하여 신뢰 제공자와 검증된 액세스 그룹을 구성할 수 있습니다. 다음 절차에 따라 Verified Access 인스턴스를 만든 다음 Verified Access에 신뢰 제공자를 연결하거나 Verified Access에서 신뢰 제공자를 분리하십시오.

주제

Verified·Access 인스턴스 생성

다음 절차에 따라 Verified·Access 인스턴스를 생성하십시오.

Verified·Access 인스턴스를 생성하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택한 다음 Verified·Access 인스턴스 생성을 선택합니다.
3. (선택 사항) 이름 및 설명에 Verified·Access 인스턴스의 이름과 설명을 입력합니다.
4. (선택 사항) 검증된 액세스를 FIPS 준수해야 하는 경우 연방 정보 프로세스 표준 (FIPS) 활성화를 선택합니다.

5. (선택 사항) 신뢰 공급자에서 Verified·Access 인스턴스에 연결할 신뢰 공급자를 선택합니다.
6. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
7. Verified·Access 인스턴스 생성을 선택합니다.

검증된 액세스 인스턴스에 신뢰 제공자를 연결합니다.

다음 절차에 따라 Verified Access 인스턴스에 신뢰 공급자를 연결하십시오.

Verified·Access 인스턴스에 신뢰 공급자를 연결하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, Verified·Access 신뢰 공급자 연결을 선택합니다.
5. Verified·Access 신뢰 공급자에서 신뢰 공급자를 선택합니다.
6. Verified·Access 신뢰 공급자 연결을 선택합니다.

검증된 액세스 인스턴스에서 신뢰 제공자를 분리합니다.

다음 절차에 따라 Verified·Access 인스턴스에서 신뢰 공급자를 분리하십시오.

Verified·Access 인스턴스에서 신뢰 공급자를 분리하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 인스턴스를 선택합니다.
4. 작업, Verified·Access 신뢰 공급자 분리를 선택합니다.
5. Verified·Access 신뢰 공급자에서 신뢰 공급자를 선택합니다.
6. Verified·Access 신뢰 공급자 분리를 선택합니다.

Verified·Access 인스턴스 삭제

Verified·Access 인스턴스를 마치면 이를 삭제할 수 있습니다. 인스턴스를 삭제하기 전에 먼저 연결된 신뢰 공급자 또는 Verified Access 그룹을 모두 제거해야 합니다.

Verified·Access 인스턴스를 삭제하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 작업, Verified·Access 인스턴스 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **`delete`**를 입력한 다음 삭제(Delete)를 선택합니다.

검증된 액세스를 다음과 통합하십시오. AWS WAF

Verified·Access에서 적용하는 인증 및 권한 부여 규칙 외에도 경계 보안을 적용할 수도 있습니다. 이렇게 하면 추가 위협으로부터 애플리케이션을 보호하는 데 도움이 될 수 있습니다. Verified Access 배포에 AWS WAF 통합하여 이 작업을 수행할 수 있습니다. AWS WAF 보호된 웹 애플리케이션 리소스에 전달되는 HTTP (S) 요청을 모니터링할 수 있는 웹 애플리케이션 방화벽입니다. 에 대한 AWS WAF자세한 내용은 AWS WAF 개발자 안내서를 참조하십시오 AWS WAF.

AWS WAF 웹 액세스 제어 목록 (ACL) 을 검증된 액세스 인스턴스에 연결하여 검증된 AWS WAF 액세스와 통합할 수 있습니다. ACL웹은 보호된 AWS WAF 리소스가 응답하는 모든 HTTP (S) 웹 요청을 세밀하게 제어할 수 있는 리소스입니다. AWS WAF 연결 또는 연결 해제 요청이 처리되는 동안 인스턴스에 연결된 모든 Verified Access 엔드포인트의 상태는 다음과 같이 표시됩니다. `updating` 요청이 완료되면 상태가 `active`(으)로 돌아갑니다. 를 사용하여 엔드포인트를 설명하면 AWS Management Console OR의 상태를 볼 수 있습니다. AWS CLI

> **Note**
>
> AWS WAF 콘솔을 사용하거나 이 통합을 수행할 수도 있습니다. API 검증된 액세스 인스턴스의 Amazon 리소스 이름 (ARN) 이 필요합니다. 다음 형식을 ARN 사용하여 이를 구성할 수 `arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}` 있습니다.

주제

- IAMVerified Access를 통합하는 데 필요한 권한 AWS WAF
- AWS WAF 웹 연결 ACL
- 통합 상태 확인 AWS WAF

- 웹 연결 끊기 AWS WAF ACL

IAMVerified Access를 통합하는 데 필요한 권한 AWS WAF

Verified AWS WAF Access와의 통합에는 작업에 직접 해당하지 않는 권한 전용 작업이 포함됩니다. API 이러한 작업은 AWS Identity and Access Management 서비스 승인 참조에 [permission only](으)로 나와 있습니다. 서비스 승인 EC2 참조에서 Amazon의 작업, 리소스 및 조건 키를 참조하십시오.

웹에서 ACL 작업하려면 AWS Identity and Access Management 주체에게 다음과 같은 권한이 있어야 합니다.

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

AWS WAF 웹 연결 ACL

다음 단계는 를 사용하여 AWS WAF 웹 액세스 제어 목록 (ACL) 을 Verified Access 인스턴스와 연결하는 방법을 보여줍니다 AWS Management Console.

> **Tip**
>
> 아래 절차를 ACL 완료하려면 기존 AWS WAF 웹이 있어야 합니다. 웹에 대한 자세한 내용은 AWS WAF 개발자 안내서의 웹 액세스 제어 목록을 ACLs 참조하십시오.

AWS WAF ACL웹을 검증된 액세스 인스턴스에 연결하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 통합 탭을 선택합니다.
5. 작업을 선택한 다음, 웹 연결을 선택합니다ACL.

6. ACL웹의 경우 기존 ACL 웹을 선택한 다음 [웹 연결] 을 선택합니다ACL.

양식을 사용하여 이 AWS Management Console 작업을 수행할 수도 있습니다. AWS WAF 자세한 내용은 개발자 안내서의 ACLAWS리소스와 웹 연결 또는 연결 해제를 참조하십시오.AWS WAF

통합 상태 확인 AWS WAF

를 사용하여 AWS WAF 웹 액세스 제어 목록 (ACL) 이 Verified Access 인스턴스와 연결되어 있는지 여부를 확인할 수 AWS Management Console있습니다.

Verified Access 인스턴스와의 AWS WAF 통합 상태를 보려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 통합 탭을 선택합니다.
5. WAF통합 상태 아래에 나열된 세부 정보를 확인하십시오. 연결 상태인 경우 상태는 웹 ACL 식별자와 함께 연결됨 또는 연결되지 않음으로 표시됩니다.

웹 연결 끊기 AWS WAF ACL

다음 단계는 를 사용하여 AWS WAF 웹 액세스 제어 목록 (ACL) 과 Verified Access 인스턴스의 연결을 끊는 방법을 보여줍니다. AWS Management Console

검증된 액세스 ACL 인스턴스에서 AWS WAF 웹 연결을 끊으려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. 통합 탭을 선택합니다.
5. 작업을 선택한 다음 웹 ACL 연결 해제를 선택합니다.
6. 웹 연결 해제를 선택하여 확인합니다. ACL

양식을 사용하여 이 AWS Management Console 작업을 수행할 수도 있습니다. AWS WAF 자세한 내용은 개발자 안내서의 ACLAWS리소스와 웹 연결 또는 연결 해제를 참조하십시오.AWS WAF

FIPS검증된 액세스에 대한 규정 준수

연방 정보 처리 표준 (FIPS) 은 민감한 정보를 보호하는 암호화 모듈에 대한 보안 요구 사항을 지정하는 미국 및 캐나다 정부 표준입니다. AWS Verified Access FIPS간행물 140-2를 준수하도록 환경을 구성하는 옵션을 제공합니다. FIPS검증된 액세스에 대한 규정 준수는 다음 AWS 지역에서 확인할 수 있습니다.

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(캘리포니아 북부)
- 미국 서부(오레곤)
- 캐나다(중부)
- AWS GovCloud (US) 서부
- AWS GovCloud (US) 동부

이 페이지에서는 신규 또는 기존 Verified Access 환경을 FIPS 규정을 준수하도록 구성하는 방법을 보여줍니다.

주제

규정 준수를 위해 FIPS 기존의 검증된 액세스 환경을 구성합니다.

기존 Verified Access 환경이 있고 FIPS 규정을 준수하도록 구성하려는 경우 규정 준수를 활성화하려면 일부 리소스를 삭제하고 다시 만들어야 합니다. FIPS

FIPS규정을 준수하도록 기존 AWS Verified Access 환경을 재구성하려면 아래 단계를 따르십시오.

1. 원래 Verified·Access 엔드포인트, 그룹 및 인스턴스를 삭제합니다. 구성된 신뢰 공급자는 재사용할 수 있습니다.
2. Verified Access 인스턴스를 생성하여 생성 시 연방 정보 프로세스 표준 (FIPS) 을 활성화해야 합니다. 또한 생성 중에 사용하려는 Verified·Access 신뢰 공급자를 드롭다운 목록에서 선택하여 연결합니다.

3. Verified·Access 그룹을 생성합니다. 그룹을 생성하는 동안 그룹을 방금 만든 Verified·Access 인스턴스와 연결합니다.
4. 하나 이상의 Verified·Access 엔드포인트를 생성합니다. 엔드포인트를 생성하는 동안 이전 단계에서 생성한 그룹과 엔드포인트를 연결합니다.

FIPS규정 준수를 위한 새로운 검증된 액세스 환경을 구성하십시오.

FIPS규정을 준수하는 새 AWS Verified Access 환경을 구성하려면 아래 단계를 따르십시오.

1. 신뢰 공급자를 구성합니다. 필요에 따라 사용자 자격 증명 신뢰 공급자와 디바이스 기반 신뢰 공급자(선택 사항)를 생성해야 합니다.
2. 검증된 액세스 인스턴스를 생성하여 프로세스 중에 연방 정보 프로세스 표준 (FIPS) 을 활성화해야 합니다. 또한 생성 중에 이전 단계에서 만든 Verified·Access 신뢰 공급자를 드롭다운 목록에서 선택하여 연결합니다.
3. Verified·Access 그룹을 생성합니다. 그룹을 생성하는 동안 그룹을 방금 만든 Verified·Access 인스턴스와 연결합니다.
4. 하나 이상의 Verified·Access 엔드포인트를 생성합니다. 엔드포인트를 생성하는 동안 이전 단계에서 생성한 그룹과 엔드포인트를 연결합니다.

Verified·Access에 대한 신뢰 공급자

신뢰 공급자는 사용자 및 장치에 대한 정보를 보내는 AWS Verified Access서비스입니다. 이러한 정보를 신뢰 컨텍스트라고 합니다. 여기에는 이메일 주소, "영업" 조직의 멤버십 등 사용자 ID에 기반한 속성이나 설치된 보안 패치, 바이러스 백신 소프트웨어 버전 등 디바이스 정보가 포함될 수 있습니다.

Verified Access는 다음 범주의 신뢰 공급자를 지원합니다.

- 사용자 자격 증명 - 사용자의 디지털 ID를 저장하고 관리하는 ID 제공업체(IdP) 서비스입니다.
- 디바이스 관리 - 노트북, 태블릿, 스마트폰과 같은 디바이스의 디바이스 관리 시스템입니다.

내용

검증된 액세스를 위한 사용자-ID 신뢰 제공자

둘 중 하나를 AWS IAM Identity Center 사용하거나 OpenID Connect와 호환되는 사용자 ID 신뢰 제공자를 사용할 수 있습니다.

내용

IAM아이덴티티 센터를 신뢰 공급자로 사용

AWS 검증된 액세스가 있는 사용자 ID 신뢰 AWS IAM Identity Center 공급자로 사용할 수 있습니다.

필수 조건 및 고려 사항

- IAMID 센터 인스턴스는 인스턴스여야 합니다. AWS Organizations 독립형 AWS 계정 IAM ID 센터 인스턴스는 작동하지 않습니다.
- 검증된 액세스 신뢰 공급자를 생성하려는 AWS 지역과 동일한 지역에서 IAM Identity Center 인스턴스를 활성화해야 합니다.

다양한 인스턴스 유형에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center의 조직 및 계정 인스턴스 관리를 참조하십시오.

IAMID 센터 신뢰 제공자 생성

AWS 계정에서 IAM ID 센터를 활성화한 후 다음 절차를 사용하여 IAM Identity Center를 검증된 액세스를 위한 신뢰 공급자로 설정할 수 있습니다.

IAMID 센터 신뢰 제공자를 만들려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 신뢰 공급자의 이름과 설명을 입력합니다.
4. 정책 참조 이름에 나중에 정책 규칙 작업 시 사용할 식별자를 입력합니다.
5. 신뢰 공급자 유형에서 사용자 신뢰 공급자를 선택합니다.
6. 사용자 신뢰 공급자 유형에서 IAM ID 센터를 선택합니다.
7. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
8. Verified·Access 신뢰 공급자 생성을 선택합니다.

IAMID 센터 신뢰 공급자를 만들려면 (AWS CLI)

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

IAMID 센터 신뢰 제공자 삭제

신뢰 공급자를 삭제하기 전에 먼저 신뢰 공급자가 연결된 인스턴스에서 모든 엔드포인트 및 그룹 구성을 제거해야 합니다.

IAMID 센터 신뢰 제공자를 삭제하려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자에서 삭제하려는 신뢰 공급자를 선택합니다.
3. 작업을 선택한 다음 Verified·Access 신뢰 공급자 삭제를 선택합니다.
4. 텍스트 상자에 `delete`를 입력하여 삭제를 확인합니다.

5. Delete(삭제)를 선택합니다.

IAMID 센터 신뢰 제공자를 삭제하려면 (AWS CLI)

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

OpenID Connect 트러스트 프로바이더 사용하기

AWS Verified Access 표준 OpenID Connect (OIDC) 메서드를 사용하는 ID 공급자를 지원합니다. OIDC호환되는 공급자를 검증된 액세스가 있는 사용자 ID 신뢰 공급자로 사용할 수 있습니다. 그러나 잠재적 OIDC 공급자가 다양하기 때문에 Verified OIDC Access와의 각 통합을 테스트할 AWS 수는 없습니다.

Verified Access는 OIDC 제공자의 신뢰 데이터를 바탕으로 평가한 신뢰 데이터를 얻습니다. `UserInfo Endpoint Scope` 파라미터는 검색할 신뢰 데이터 집합을 결정하는 데 사용됩니다. 신뢰 데이터를 수신한 후에는 이를 기준으로 Verified·Access 정책이 평가됩니다.

> **Note**
>
> Verified Access는 Verified Access 정책을 평가할 때 OIDC 공급자가 `ID token` 보낸 신뢰 데이터를 사용하지 않습니다. `UserInfo Endpoint`의 신뢰 데이터만 정책에 따라 평가됩니다.

내용

- 신뢰 제공자를 OIDC 만들기 위한 사전 요구 사항
- OIDC신뢰 제공자 생성
- OIDC신뢰 제공자 수정
- OIDC신뢰 제공자 삭제

신뢰 제공자를 OIDC 만들기 위한 사전 요구 사항

신뢰 공급자 서비스에서 직접 다음 정보를 수집해야 합니다.

- Issuer
- 권한 부여 엔드포인트
- Token 엔드포인트

- UserInfo 엔드포인트
- 클라이언트 ID
- 클라이언트 암호
- 범위

OIDC신뢰 제공자 생성

다음 절차를 사용하여 AS를 신뢰 OIDC 공급자로 만드십시오.

OIDC신뢰 제공자를 만들려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 신뢰 공급자의 이름과 설명을 입력합니다.
4. 정책 참조 이름에 나중에 정책 규칙 작업 시 사용할 식별자를 입력합니다.
5. 신뢰 공급자 유형에서 사용자 신뢰 공급자를 선택합니다.
6. 사용자 신뢰 제공자 유형에서 OIDC (OpenID Connect) 를 선택합니다.
7. 발급자의 경우 발급자의 OIDC 식별자를 입력합니다.
8. 권한 부여 엔드포인트에 권한 부여 엔드포인트 전체를 URL 입력합니다.
9. 토큰 엔드포인트에 전체 URL 토큰 엔드포인트를 입력합니다.
10. 사용자 엔드포인트에 전체 URL 사용자 엔드포인트를 입력합니다.
11. 클라이언트 ID에 OAuth 2.0 클라이언트 식별자를 입력합니다.
12. 클라이언트 암호에 OAuth 2.0 클라이언트 암호를 입력합니다.
13. 자격 증명 공급자가 정의한 공백으로 구분된 범위 목록을 입력합니다. 범위에는 최소한 “openid” 범위가 필요합니다.
14. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
15. Verified·Access 신뢰 공급자 생성을 선택합니다.

> **Note**
>
> OIDC제공자의 허용 목록에 URI 리디렉션을 추가해야 합니다. 이 용도로는 Verified·Access 엔드포인트의 `ApplicationDomain`를 사용하는 것이 좋습니다. 이 정보는 Verified AWS

Management Console Access 엔드포인트의 세부 정보 탭 아래에 있거나 를 사용하여 엔드포인트를 AWS CLI 설명하여 찾을 수 있습니다. OIDC제공자의 허용 목록에 다음을 추가하십시오. https:///oauth2/idpresponse `ApplicationDomain`

신뢰 OIDC 제공자를AWS CLI 만들려면 ()

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

OIDC신뢰 제공자 수정

신뢰 공급자 생성 후 해당 구성을 업데이트할 수 있습니다.

OIDC신뢰 제공자를 수정하려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자에서 수정하려는 신뢰 공급자를 선택합니다.
3. 작업을 선택한 다음 Verified·Access 신뢰 공급자 수정을 선택합니다.
4. 변경할 옵션을 수정합니다.
5. Verified·Access 신뢰 공급자 수정을 선택합니다.

OIDC신뢰 제공자를 수정하려면 (AWS CLI)

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

OIDC신뢰 제공자 삭제

사용자 신뢰 공급자를 삭제하기 전에 먼저 신뢰 공급자가 연결된 인스턴스에서 모든 엔드포인트 및 그룹 구성을 제거해야 합니다.

OIDC신뢰 제공자를 삭제하려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자에서 삭제하려는 신뢰 공급자를 선택합니다.
3. 작업을 선택한 다음 Verified·Access 신뢰 공급자 삭제를 선택합니다.

4. 텍스트 상자에 delete를 입력하여 삭제를 확인합니다.
5. Delete(삭제)를 선택합니다.

OIDC신뢰 제공자를 삭제하려면 (AWS CLI)

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

검증된 액세스를 위한 장치 기반 신뢰 제공자

검증된 액세스와 함께 AWS 장치 신뢰 공급자를 사용할 수 있습니다. Verified Access 인스턴스를 통해 여러 디바이스 신뢰 공급자를 사용할 수 있습니다.

내용

- 지원되는 디바이스 신뢰 공급자
- 디바이스 기반 신뢰 공급자 생성
- 디바이스 기반 신뢰 공급자 수정
- 디바이스 기반 신뢰 공급자 삭제

지원되는 디바이스 신뢰 공급자

다음 디바이스 신뢰 공급자는 Verified Access와 통합될 수 있습니다.

- CrowdStrike — 검증된 액세스를 통한 사설 애플리케이션 보호 CrowdStrike
- Jamf - Verified·Access를 Jamf 디바이스 자격 증명과 통합
- JumpCloud — 통합 JumpCloud 및 AWS 검증된 액세스

디바이스 기반 신뢰 공급자 생성

다음 단계에 따라 Verified·Access와 함께 사용할 디바이스 신뢰 공급자를 생성하고 구성하십시오.

검증된 액세스 장치 신뢰 제공자 (AWS 콘솔) 를 만들려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택한 다음 Verified·Access 신뢰 공급자 생성을 선택합니다.

3. (선택 사항) 이름 태그 및 설명에 신뢰 공급자의 이름과 설명을 입력합니다.
4. 나중에 정책 참조 이름에 대한 정책 규칙을 작업할 때 사용할 식별자를 입력합니다.
5. 신뢰 공급자 유형에서 디바이스 자격 증명을 선택합니다.
6. 디바이스 ID 유형에서 Jamf CrowdStrike, 또는 JumpCloud를 선택합니다.
7. 테넌트 ID에는 테넌트 애플리케이션의 식별자를 입력합니다.
8. (선택 사항) 공개 서명 키의 URL 경우 장치 신뢰 공급자가 URL 공유하는 고유 키를 입력합니다. (Jamf CrowdStrike 또는 Jumpcloud에는 이 매개변수가 필요하지 않습니다.)
9. Verified·Access 신뢰 공급자 생성을 선택합니다.

> **Note**
>
> OIDC제공자의 허용 목록에 URI 리디렉션을 추가해야 합니다. 이 용도로는 Verified·Access 엔드포인트의 `DeviceValidationDomain`를 사용하는 것이 좋습니다. 이 정보는 Verified AWS Management Console Access 엔드포인트의 세부 정보 탭 아래에 있거나 를 사용하여 엔드포인트를 AWS CLI 설명하여 찾을 수 있습니다. OIDC제공자의 허용 목록에 다음을 추가하십시오. https:///oauth2/idpresponse `DeviceValidationDomain`

검증된 액세스 장치 신뢰 제공자를 만들려면 ()AWS CLI

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

디바이스 기반 신뢰 공급자 수정

신뢰 공급자 생성 후 해당 구성을 업데이트할 수 있습니다.

검증된 액세스 장치 신뢰 제공자를 수정하려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택합니다.
3. 신뢰 공급자를 선택합니다.
4. 작업을 선택한 다음 Verified·Access 신뢰 공급자 수정을 선택합니다.
5. 필요에 따라 설명을 수정합니다.

6. (선택 사항) 공개 서명 키의 URL 경우 장치 신뢰 공급자가 URL 공유하는 고유 키를 수정하십시오. (장치 신뢰 공급자가 Jamf CrowdStrike 또는 Jumpcloud인 경우에는 이 매개변수가 필요하지 않습니다.)
7. Verified·Access 신뢰 공급자 수정을 선택합니다.

검증된 액세스 장치 신뢰 제공자를 수정하려면 ()AWS CLI

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

디바이스 기반 신뢰 공급자 삭제

신뢰 공급자 사용을 마치면 이를 삭제할 수 있습니다.

검증된 액세스 장치 신뢰 제공자를 삭제하려면 (AWS 콘솔)

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 신뢰 공급자를 선택합니다.
3. Verified·Access 신뢰 공급자에서 삭제하려는 신뢰 공급자를 선택합니다.
4. 작업을 선택한 다음 Verified·Access 신뢰 공급자 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **delete**를 입력한 다음 삭제(Delete)를 선택합니다.

검증된 액세스 장치 신뢰 제공자를 삭제하려면 (AWS CLI)

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

Verified·Access 그룹

AWS Verified Access 그룹은 검증된 액세스 엔드포인트와 그룹 수준의 검증된 액세스 정책의 모음입니다. 그룹 내 각 엔드포인트는 Verified·Access 정책을 공유합니다. 그룹을 사용하여 공통 보안 요구 사항이 있는 엔드포인트를 한데 모을 수 있습니다. 이렇게 하면 여러 애플리케이션의 보안 요구 사항에 대해 하나의 정책을 사용하여 정책 관리를 단순화할 수 있습니다.

예를 들어 모든 판매 애플리케이션을 그룹화하고 그룹 전체 액세스 정책을 설정할 수 있습니다. 그런 다음 이 정책을 사용하여 모든 판매 애플리케이션에 대해 하나의 공통된 최소 보안 요구 사항 집합을 정의할 수 있습니다. 이 접근 방식은 정책 관리를 단순화하는 데 도움이 됩니다.

그룹을 생성할 때 Verified·Access 인스턴스와 그룹을 연결해야 합니다. 엔드포인트를 생성하는 과정에서 엔드포인트를 그룹과 연결합니다.

Tasks

- Verified·Access 그룹 생성
- Verified·Access 그룹 정책 수정
- Verified·Access 그룹 삭제

Verified·Access 그룹 생성

Verified·Access 그룹을 생성하려면 다음 절차를 따르십시오.

Verified·Access 그룹을 생성하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 그룹을 선택한 다음 Verified·Access 그룹 생성을 선택합니다.
3. (선택 사항) 이름 태그 및 설명에 그룹의 이름과 설명을 입력합니다.
4. Verified·Access 인스턴스의 경우 그룹과 연결할 Verified·Access 인스턴스를 선택합니다.
5. (선택 사항) 정책 정의에는 그룹에 적용할 Verified·Access 정책을 입력합니다.
6. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
7. Verified·Access 그룹 생성을 선택합니다.

Verified·Access 그룹 정책 수정

다음 절차에 따라 Verified·Access 그룹 정책을 수정하십시오.

Verified·Access 그룹 정책을 수정하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 그룹을 선택한 다음 수정하려는 정책을 가진 그룹을 선택합니다.
3. 작업을 선택한 다음 Verified·Access 그룹 정책 수정을 선택합니다.
4. (선택 사항) 현재 목표에 따라 활성화 정책을 켜거나 끕니다.
5. (선택 사항) 정책에 그룹에 적용할 Verified·Access 정책을 입력합니다.
6. Verified·Access 그룹 정책 수정을 선택합니다.

Verified·Access 그룹 삭제

Verified·Access 그룹 사용을 마치면 이를 삭제할 수 있습니다.

Verified·Access 그룹을 삭제하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 그룹을 선택합니다.
3. 그룹을 선택합니다.
4. 작업, Verified·Access 그룹 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **`delete`**를 입력한 다음 삭제(Delete)를 선택합니다.

Verified·Access 엔드포인트

Verified·Access 엔드포인트는 애플리케이션을 나타냅니다. 각 엔드포인트는 Verified Access 그룹과 연결되며 그룹에 대한 액세스 정책을 상속합니다. 필요에 따라 애플리케이션별 엔드포인트 정책을 각 엔드포인트에 연결할 수 있습니다.

내용

Verified·Access 엔드포인트 유형

가능한 검증된 액세스 엔드포인트 유형은 다음과 같습니다.

- 로드 밸런서 – 애플리케이션 요청이 로드 밸런서로 전송되어 애플리케이션에 배포됩니다.
- 네트워크 인터페이스 - 애플리케이션 요청은 지정된 프로토콜 및 포트를 사용하여 네트워크 인터페이스로 전송됩니다.

검증된 액세스가 공유 VPCs 및 서브넷에서 작동하는 방식

공유 VPC 서브넷과 관련된 동작은 다음과 같습니다.

- 검증된 액세스 엔드포인트는 VPC 서브넷 공유를 통해 지원됩니다. 참여자는 공유 서브넷에서 Verified Access 엔드포인트를 생성할 수 있습니다.
- 엔드포인트를 생성한 참여자가 엔드포인트 소유자이며, 엔드포인트를 수정할 수 있는 유일한 당사자가 됩니다. VPC소유자는 엔드포인트를 수정할 수 없습니다.
- Verified Access 엔드포인트는 AWS 로컬 영역에 생성할 수 없으므로 Local Zones를 통한 공유는 불가능합니다.

자세한 내용은 Amazon VPC 사용 설명서의 다른 VPC 계정과 공유를 참조하십시오.

Verified Acces에 대한 로드 밸런서 엔드포인트 생성

다음 절차를 사용하여 검증된 액세스를 위한 로드 밸런서 엔드포인트를 만들 수 있습니다. 로드 밸런서에 대한 자세한 내용은 Elastic Load Balancing 사용 설명서를 참조하십시오.

요구 사항

- IPv4트래픽만 지원됩니다.
- HTTP및 HTTPS 프로토콜만 지원됩니다.
- 로드 밸런서는 Application Load Balancer 또는 Network Load Balancer여야 하며, 내부 로드 밸런서여야 합니다.
- 로드 밸런서와 서브넷은 동일한 가상 사설 클라우드 () VPC 에 속해야 합니다.
- HTTPS로드 밸런서는 자체 서명 인증서 또는 공개 인증서를 사용할 수 있습니다. TLS
- 애플리케이션에 사용할 도메인 이름을 제공해야 합니다. 이 DNS 이름은 사용자가 애플리케이션에 액세스하는 데 사용할 공개 이름입니다. 또한 이 도메인 이름과 일치하는 CN이 포함된 공개 SSL 인증서를 제공해야 합니다. 를 사용하여 인증서를 만들거나 가져올 수 AWS Certificate Manager있습니다.

로드 밸런서 엔드포인트를 생성하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. Verified·Access 엔드포인트 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
5. Verified·Access 그룹에서 엔드포인트의 Verified·Access 그룹을 선택합니다.
6. 애플리케이션 세부 정보를 보려면 다음을 수행하십시오.
   a. 애플리케이션 도메인의 경우 애플리케이션 DNS 이름을 입력합니다.
   b. 도메인 인증서에서 ARN 공개 TLS 인증서를 선택합니다.
7. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 첨부 유형에서 선택합니다 VPC.
   b. 보안 그룹(Security groups)에서 엔드포인트의 보안 그룹을 선택합니다. 로드 밸런서로 들어오는 Verified·Access 엔드포인트의 트래픽은 이 보안 그룹과 연결됩니다.

c. 엔드포인트 도메인 접두사의 경우 Verified Access가 엔드포인트에 대해 생성하는 DNS 이름 앞에 사용자 지정 식별자를 입력합니다.

d. 엔드포인트 유형에서 로드 밸런서를 선택합니다.

e. 프로토콜의 경우 또는 를 선택합니다. HTTPSHTTP

f. 포트에 포트 번호를 입력합니다.

g. 로드 밸런서에서 로드 ARN 밸런서를 선택합니다.

h. 서브넷에서 로드 밸런서에 대한 서브넷을 선택합니다.

8. (선택 사항) 정책 정의에는 엔드포인트에 대한 Verified·Access 정책을 입력합니다.
9. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
10. Verified·Access 엔드포인트 생성을 선택합니다.

Verified·Access에 대한 네트워크 인터페이스 엔드포인트 생성

다음 절차에 따라 네트워크 인터페이스 엔드포인트를 생성하십시오.

요구 사항

- IPv4트래픽만 지원됩니다.
- HTTP및 HTTPS 프로토콜만 지원됩니다.
- 네트워크 인터페이스는 보안 그룹과 동일한 가상 사설 클라우드 (VPC) 에 속해야 합니다.
- 네트워크 인터페이스의 프라이빗 IP를 사용하여 트래픽을 전달합니다.
- 애플리케이션에 사용할 도메인 이름을 제공해야 합니다. 이 DNS 이름은 사용자가 애플리케이션에 액세스하는 데 사용할 퍼블릭 이름입니다. 또한 이 도메인 이름과 일치하는 CN이 포함된 공개 SSL 인증서를 제공해야 합니다. 를 사용하여 인증서를 만들거나 가져올 수 AWS Certificate Manager있습니다.

네트워크 인터페이스 엔드포인트를 생성하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. Verified·Access 엔드포인트 생성을 선택합니다.
4. (선택 사항) 이름 태그 및 설명에 엔드포인트의 이름과 설명을 입력합니다.
5. Verified·Access 그룹에서 엔드포인트의 Verified·Access 그룹을 선택합니다.

6. 애플리케이션 세부 정보를 보려면 다음을 수행하십시오.
   a. 애플리케이션 도메인의 경우 애플리케이션 DNS 이름을 입력합니다.
   b. 도메인 인증서에서 ARN 공개 TLS 인증서를 선택합니다.
7. 엔드포인트 세부 정보에서 다음을 수행합니다.
   a. 첨부 유형에서 선택합니다 VPC.
   b. 보안 그룹(Security groups)에서 엔드포인트의 보안 그룹을 선택합니다. 네트워크 인터페이스로 들어오는 Verified·Access 엔드포인트의 트래픽은 이 보안 그룹과 연결됩니다.
   c. 엔드포인트 도메인 접두사의 경우 Verified Access가 엔드포인트에 대해 생성하는 DNS 이름 앞에 사용자 지정 식별자를 입력합니다.
   d. Endpoint type(엔드포인트 유형)에서 Network interface(네트워크 인터페이스)를 선택합니다.
   e. 프로토콜의 경우 또는 를 선택합니다. HTTPSHTTP
   f. 포트에 포트 번호를 입력합니다.
   g. 네트워크 인터페이스에서 네트워크 인터페이스를 선택합니다.
8. (선택 사항) 정책 정의에는 엔드포인트에 대한 Verified·Access 정책을 입력합니다.
9. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
10. Verified·Access 엔드포인트 생성을 선택합니다.

Verified·Access 엔드포인트에서 발생하는 트래픽 허용

Verified Access 엔드포인트에서 발생하는 트래픽을 허용하도록 애플리케이션의 보안 그룹을 구성할 수 있습니다. 엔드포인트의 보안 그룹을 소스로 지정하는 인바운드 규칙을 추가하면 됩니다. 애플리케이션이 Verified Access 엔드포인트로부터만 트래픽을 수신하도록 추가 인바운드 규칙을 제거하는 것이 좋습니다.

기존 아웃바운드 규칙을 유지하는 것이 좋습니다.

애플리케이션의 보안 그룹 규칙을 업데이트하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 검증된 액세스 엔드포인트를 선택하고 세부 정보 탭에서 보안 그룹을 IDs 찾은 다음 엔드포인트의 보안 그룹 ID를 복사합니다.
4. 탐색 창에서 보안 그룹을 선택합니다.

5. 대상과 관련된 보안 그룹에 대한 상자를 선택한 다음 작업, 인바운드 규칙 편집을 선택합니다.
6. Verified·Access 엔드포인트에서 발생하는 트래픽을 허용하는 보안 그룹 규칙을 추가하려면 다음을 수행하십시오.
   a. 규칙 추가를 선택합니다.
   b. 유형에서 모든 트래픽 또는 허용할 특정 트래픽을 선택합니다.
   c. 소스로 사용자 지정을 선택하고 엔드포인트에 보안 그룹의 ID를 붙여넣습니다.
7. (선택 사항) 트래픽이 Verified Access 엔드포인트에서만 발생하도록 하려면 다른 인바운드 보안 그룹 규칙을 삭제합니다.
8. 규칙 저장을 선택합니다.

Verified·Access 엔드포인트 수정

검증된 액세스 엔드포인트를 생성한 후 해당 구성을 수정할 수 있습니다.

Verified·Access 엔드포인트를 수정하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 엔드포인트를 선택합니다.
4. 작업, Verified·Access 엔드포인트 수정을 선택합니다.
5. 필요에 따라 엔드포인트 세부 사항을 수정합니다.
6. Verified·Access 엔드포인트 수정을 선택합니다.

Verified·Access 엔드포인트 정책 수정

Verified·Access 엔드포인트를 생성한 후 해당 정책을 수정할 수 있습니다.

Verified·Access 정책을 수정하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 수정하려는 정책이 있는 엔드포인트를 선택합니다.
4. 작업, Verified·Access 엔드포인트 정책 수정을 선택합니다.

5. (선택 사항) 현재 목표에 따라 활성화 정책을 켜거나 끕니다.
6. (선택 사항) 정책에는 엔드포인트에 적용할 Verified·Access 정책을 입력합니다.
7. Verified·Access 엔드포인트 정책 수정을 선택합니다.

Verified·Access 엔드포인트 삭제

Verified·Access 엔드포인트 사용을 마치면 엔드포인트를 삭제할 수 있습니다.

Verified·Access 엔드포인트를 삭제하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 엔드포인트를 선택합니다.
3. 엔드포인트를 선택합니다.
4. 작업, Verified·Access 엔드포인트 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **delete**을 입력한 다음 삭제를 선택합니다.

신뢰 제공자가 Verified Access로 보낸 신뢰 데이터

신뢰 데이터는 신뢰 제공자가 보내는 AWS Verified Access 데이터입니다. 신뢰 데이터는 “사용자 클레임” 또는 “신뢰 컨텍스트”라고도 합니다. 데이터에는 일반적으로 사용자 또는 디바이스에 대한 정보가 포함됩니다. 신뢰 데이터의 예로는 사용자 이메일, 그룹 구성원, 장치 운영 체제 버전, 장치 보안 상태 등이 있습니다. 전송되는 정보는 신뢰 제공자에 따라 다르므로 전체 및 업데이트된 신뢰 데이터 목록은 신뢰 제공자의 설명서를 참조해야 합니다.

그러나 Verified·Access 로깅 기능을 사용하면 신뢰 공급자로부터 어떤 신뢰 데이터가 전송되고 있는지도 확인할 수 있습니다. 이는 응용 프로그램에 대한 액세스를 허용하거나 거부하는 정책을 정의할 때 유용할 수 있습니다. 로그에 신뢰 컨텍스트를 포함하는 방법에 대한 정보는 검증된 액세스 신뢰 컨텍스트를 활성화 또는 비활성화합니다.를 참조하십시오.

이 섹션에는 정책 작성을 시작하는 데 도움이 되는 샘플 신뢰 데이터와 예제가 포함되어 있습니다. 여기에 제공된 정보는 설명을 위한 용도로만 사용되며 공식적인 참고 자료는 아닙니다.

내용

- Verified Access 신뢰 데이터의 기본 컨텍스트
- AWS IAM Identity Center 검증된 액세스 신뢰 데이터의 컨텍스트
- 검증된 액세스 신뢰 데이터에 대한 타사 신뢰 제공자 컨텍스트
- 사용자는 Verified Access에서 통과 및 서명 확인을 요구합니다.

Verified Access 신뢰 데이터의 기본 컨텍스트

AWS Verified Access 구성된 신뢰 공급자에 관계없이 모든 Cedar 평가에 기본적으로 현재 HTTP 요청에 대한 일부 요소를 포함합니다. 정책을 평가할 때 Verified Access는 Cedar 컨텍스트의 현재 HTTP 요청에 대한 데이터를 아래에 포함합니다. `context.http_request` key 원하는 경우 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON스키마는 평가에 포함되는 데이터를 보여줍니다.

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

다음은 HTTP 요청 데이터를 기준으로 평가하는 정책의 예입니다.

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

AWS IAM Identity Center 검증된 액세스 신뢰 데이터의 컨텍스트

정책을 평가할 때 신뢰 AWS IAM Identity Center 제공자로 정의하는 경우 Cedar 컨텍스트의 신뢰 데이터를 신뢰 제공자 구성에서 “정책 참조 이름”으로 지정한 키 아래에 AWS Verified Access 포함시킵니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다.

> **Note**
>
> 신뢰 공급자의 컨텍스트 키는 신뢰 공급자를 생성할 때 구성한 정책 참조 이름에서 가져옵니다. 예를 들어, 정책 참조 이름을 “idp123"으로 구성하면 컨텍스트 키는 “context.idp123"이 됩니다. 정책을 생성할 때 올바른 컨텍스트 키를 사용하고 있는지 확인합니다.

다음 JSON스키마는 평가에 포함되는 데이터를 보여줍니다.

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

다음은 에서 제공한 AWS IAM Identity Center신뢰 데이터를 기준으로 평가하는 정책의 예입니다.

```
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**
>
> 그룹 이름을 변경할 수 있으므로 IAM Identity Center는 해당 그룹 ID를 사용하는 그룹을 참조합니다. 이렇게 하면 그룹 이름을 변경할 때 정책 설명이 위반되는 것을 방지할 수 있습니다.

검증된 액세스 신뢰 데이터에 대한 타사 신뢰 제공자 컨텍스트

이 섹션에서는 타사 신뢰 제공자가 AWS Verified Access 제공하는 신뢰 데이터에 대해 설명합니다.

> **Note**
>
> 신뢰 공급자의 컨텍스트 키는 신뢰 공급자를 생성할 때 구성한 정책 참조 이름에서 가져옵니다. 예를 들어, 정책 참조 이름을 “idp123"으로 구성하면 컨텍스트 키는 “context.idp123"이 됩니다. 정책을 생성할 때 올바른 컨텍스트 키를 사용하고 있는지 확인합니다.

내용

브라우저 확장

장치 신뢰 컨텍스트를 액세스 정책에 통합하려는 경우 AWS Verified Access 브라우저 확장 프로그램이나 다른 파트너의 브라우저 확장이 필요합니다. Verified·Access는 현재 Google Chrome과 Mozilla Firefox 브라우저를 지원합니다.

현재 Jamf (macOS 장치 지원), Windows 11 및 Windows 10 장치 지원) CrowdStrike , 그리고 (Windows 및 macOS 모두 지원) 의 세 가지 장치 신뢰 공급자를 지원합니다. JumpCloud

- 정책에 Jamf trust 데이터를 사용하는 경우 사용자는 Chrome 웹 스토어 또는 Firefox 애드온 사이트에서 AWS Verified Access 브라우저 확장 프로그램을 다운로드하여 장치의 설치해야 합니다.
- 정책에 CrowdStrike신뢰 데이터를 사용하는 경우 먼저 사용자가 AWS Verified Access 기본 메시징 호스트 (직접 다운로드 링크) 를 설치해야 합니다. 이 구성 요소는 사용자 장치에서 실행되는 CrowdStrike 에이전트로부터 신뢰 데이터를 가져오는 데 필요합니다. 그런 다음 이 구성 요소를 설치한 후 사용자는 Chrome 웹 스토어 또는 Firefox 애드온 사이트의 AWS Verified Access 브라우저 확장 프로그램을 장치에 설치해야 합니다.
- 를 사용하는 JumpCloud경우 사용자는 Chrome 웹 스토어 또는 Firefox 애드온 사이트의 JumpCloud 브라우저 확장 프로그램을 장치에 설치해야 합니다.

Jamf

Jamf는 타사 신뢰 공급자입니다. 정책을 평가할 때 Jamf를 신뢰 공급자로 정의하면 Verified·Access는 신뢰 공급자 구성에서 “정책 참조 이름(Policy Reference Name)”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함시킵니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON스키마는 평가에 포함되는 데이터를 보여줍니다.

검증된 액세스와 함께 Jamf를 사용하는 방법에 대한 자세한 내용은 Jamf 웹 사이트의 AWS 검증된 액세스와 Jamf 장치 ID 통합을 참조하십시오.

```
{
    "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)"
        }
```

```
    }
}
```

다음은 Jamf에서 제공하는 신뢰 데이터를 기준으로 평가하는 정책의 예입니다.

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

Cedar는 Jamf의 위험 점수와 같은 Enum을 처리하는 데 유용한 `.contains()` 함수를 제공합니다.

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

CrowdStrike

CrowdStrike 타사 신뢰 제공업체입니다. 정책을 평가할 때 신뢰 CrowdStrike 제공자로 정의하면 Verified Access는 사용자가 신뢰 제공자 구성에서 “정책 참조 이름”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함합니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON스키마는 평가에 포함되는 데이터를 보여줍니다.

검증된 CrowdStrike 액세스와 함께 사용하는 방법에 대한 자세한 내용은 GitHub 웹 사이트를 통한 개인 응용 프로그램 보안을 참조하십시오. CrowdStrike AWS Verified Access

```
{
  "title": "CrowdStrike device data specification",
  "type": "object",
  "properties": {
    "assessment": {
      "type": "object",
      "description": "Data about CrowdStrike's assessment of the device",
      "properties": {
        "overall": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts as a weighted
 average of the OS and and Sensor Config scores"
        },
        "os": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the OS-
specific settings monitored on the host"
```

```
        },
        "sensor_config": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the
different sensor policies monitored on the host"
        },
        "version": {
          "type": "string",
          "description": "The version of the scoring algorithm being used"
        }
      }
    },
    "cid": {
      "type": "string",
      "description": "Customer ID (CID) unique to the customer's environemnt"
    },
    "exp": {
      "type": "integer",
      "description": "unixtime, The expiration time of the token"
    },
    "iat": {
      "type": "integer",
      "description": "unixtime, The issued time of the token"
    },
    "jwk_url": {
      "type": "string",
      "description": "URL that details the JWT signing"
    },
    "platform": {
      "type": "string",
      "enum": ["Windows 10", "Windows 11", "macOS"],
      "description": "Operating system of the endpoint"
    },
    "serial_number": {
      "type": "string",
      "description": "The serial number of the device derived by unique system
information"
    },
    "sub": {
      "type": "string",
      "description": "Unique CrowdStrike Agent ID (AID) of machine"
    },
    "typ": {
      "type": "string",
```

```
      "enum": ["crowdstrike-zta+jwt"],
      "description": "Generic name for this JWT media. Client MUST reject any other
 type"
    }
  }
}
```

다음은 에서 제공한 CrowdStrike 신뢰 데이터를 기준으로 평가하는 정책의 예입니다.

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

JumpCloud

JumpCloud 타사 신뢰 공급자입니다. 정책을 평가할 때 신뢰 JumpCloud 제공자로 정의하면 Verified Access는 사용자가 신뢰 제공자 구성에서 “정책 참조 이름”으로 지정한 키 아래에 Cedar 컨텍스트의 신뢰 데이터를 포함합니다. 원하는 경우 신뢰 데이터를 기준으로 평가하는 정책을 작성할 수 있습니다. 다음 JSON스키마는 평가에 포함되는 데이터를 보여줍니다.

AWS 검증된 JumpCloud 액세스와 함께 사용하는 방법에 대한 자세한 내용은 JumpCloud 웹 사이트의 통합 JumpCloud 및 AWS 검증된 액세스 설정을 참조하십시오.

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

다음은 에서 제공하는 신뢰 컨텍스트를 기준으로 평가하는 정책의 예입니다. JumpCloud

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

사용자는 Verified Access에서 통과 및 서명 확인을 요구합니다.

AWS Verified Access 인스턴스가 사용자를 성공적으로 인증한 후 IdP로부터 받은 사용자 클레임을 Verified Access 엔드포인트로 전송합니다. 사용자 클레임에 서명되므로 애플리케이션은 서명을 확인하고 Verified Access에서 클레임을 전송했는지도 확인할 수 있습니다. 이 프로세스 중에 다음 HTTP 헤더가 추가됩니다.

`x-amzn-ava-user-context`

이 헤더에는 JSON 웹 토큰 (JWT) 형식의 사용자 클레임이 포함되어 있습니다. JWT형식에는 URL base64로 인코딩된 헤더, 페이로드 및 서명이 포함됩니다. Verified Access는 ES384 (SHA-384 해시 알고리즘을 사용하는 ECDSA 서명 알고리즘) 을 사용하여 서명을 생성합니다. JWT

애플리케이션은 이러한 클레임을 개인화 또는 기타 사용자별 경험에 사용할 수 있습니다. 애플리케이션 개발자는 사용하기 전에 자격 증명 공급자가 제공하는 각 클레임의 고유성 및 검증 수준에 대해 스스로 학습해야 합니다. 일반적으로 sub 클레임은 특정 사용자를 식별하는 가장 좋은 방법입니다.

내용

예: 사용자 클레임을 위한 서명 JWT OIDC

다음 예시는 OIDC 사용자 클레임의 헤더와 페이로드가 JWT 형식에서 어떻게 보일지 보여줍니다.

헤더의 예:

```
{
"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)
}
```

페이로드의 예:

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

```
}
```

예: IAM Identity Center 사용자 클레임을 JWT 위한 서명

다음 예는 IAM Identity Center 사용자 클레임의 헤더 및 페이로드가 JWT 형식에서 어떻게 표시되는지 보여줍니다.

> **Note**
>
> IAMIdentity Center의 경우 사용자 정보만 클레임에 포함됩니다.

헤더의 예:

```
{
"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)
}
```

페이로드의 예:

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

퍼블릭 키

Verified Access 인스턴스는 사용자 클레임을 암호화하지 않으므로 Verified Access 엔드포인트를 사용하도록 구성하는 것이 좋습니다. HTTPS Verified Access 엔드포인트를 사용하도록 HTTP 구성한 경우 보안 그룹을 사용하여 엔드포인트로 향하는 트래픽을 제한해야 합니다.

클레임을 기반으로 권한 부여를 수행하기 전에 서명을 확인하는 것이 좋습니다. 퍼블릭 키를 가져오려면 JWT 헤더에서 키 ID를 가져오고 이를 사용하여 엔드포인트에서 퍼블릭 키를 조회하십시오. 각 AWS 리전 엔드포인트는 다음과 같습니다.

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

예: 검색 및 디코딩 JWT

다음 코드 예에서는 Python 3.9로 키 ID, 퍼블릭 키 및 페이로드를 가져오는 방법을 보여줍니다.

```
import jwt
import requests
import base64
import json

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

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

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

Verified·Access 정책

AWS Verified Access 정책을 통해 에서 호스팅되는 애플리케이션에 액세스하기 위한 규칙을 정의할 수 AWS있습니다. AWS 정책 언어인 Cedar로 작성되었습니다. Cedar를 사용하면 Verified·Access와 함께 사용하도록 구성한 ID 또는 디바이스 기반 신뢰 공급자로부터 전송된 신뢰 데이터를 기준으로 평가되는 정책을 생성할 수 있습니다.

Cedar 정책 언어에 대한 더 자세한 내용은 Cedar 참조 안내서를 참조하십시오.

이 섹션에서는 Verified Access 정책의 구조, 포함된 내용, 정의 방법을 설명하고 몇 가지 예를 제공합니다.

내용

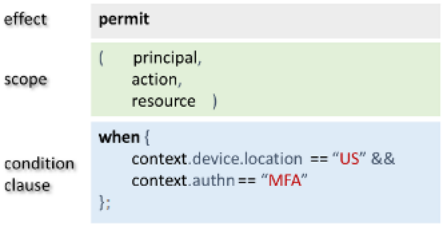
Verified·Access의 정책 작업

Verified·Access 그룹을 생성하거나 Verified·Access 엔드포인트를 생성할 때 Verified·Access 정책을 정의할 수 있는 옵션이 있습니다. Verified Access 정책을 정의하지 않고 그룹 또는 엔드포인트를 생성할 수 있지만 정책을 정의할 때까지 모든 액세스 요청이 차단됩니다.

기존 Verified·Access 그룹 또는 엔드포인트를 생성한 후 정책을 추가하거나 변경하려면 Verified·Access 그룹 정책 수정 또는 Verified·Access 엔드포인트 정책 수정를(을) 참조하십시오.

검증된 액세스 정책 설명 구조

이 섹션에서는 AWS Verified Access 정책 설명과 평가 방법을 설명합니다. 단일 Verified·Access 정책에 여러 설명을 포함할 수 있습니다. 다음 다이어그램은 Verified·Access 정책의 구조를 보여줍니다.



정책에는 다음 부분이 포함됩니다.

- 효과 – 정책 설명이 `permit` (`Allow`) 또는 `forbid` (`Deny`).인지 지정합니다.
- 범위 – 효과가 적용되는 주체, 작업 및 리소스를 지정합니다. 이전 예와 같이 특정 주체, 작업 또는 리소스를 식별하지 않음으로써 Cedar의 범위를 정의하지 않은 상태로 둘 수 있습니다. 이 경우 가능한 모든 주체, 작업 및 리소스에 정책이 적용됩니다.
- 조건 조항 - 효과가 적용되는 컨텍스트를 지정합니다.

> **Important**
>
> Verified Access의 경우 조건 조항의 신뢰 데이터를 참조하여 정책을 완전히 표현합니다. 정책 범위는 항상 정의되지 않은 상태로 유지해야 합니다. 그런 다음 조건 조항에서 ID 및 디바이스 신뢰 컨텍스트를 사용하여 액세스를 지정할 수 있습니다.

간단한 정책 예제

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

위 예제에서는 && 연산자를 사용하여 정책 설명에 조건 조항을 두 개 이상 사용할 수 있습니다. Cedar 정책 언어를 사용하면 사용자 맞춤의 세밀하고 광범위한 정책 설명을 작성할 수 있는 표현력을 얻을 수 있습니다. 추가 예제는 다음(검증된 액세스 예제 정책)을 참조하십시오.

검증된 액세스 정책 평가

정책 문서는 하나 이상의 정책 설명(`permit` 또는 `forbid` 설명)의 집합입니다. 정책은 조건부 조항(`when` 진술)이 참일 경우 적용됩니다. 정책 문서에 액세스를 허용하려면 문서에 있는 하나 이상의 허

가 정책이 적용되어야 하며 금지 정책은 적용할 수 없습니다. 허가 정책이 적용되지 않고/않거나 하나 이상의 금지 정책이 적용되는 경우에는 정책 문서가 액세스를 거부합니다. Verified Access 그룹과 Verified Access 엔드포인트 모두에 대해 정책 문서를 정의한 경우 두 문서 모두 액세스를 허용해야 합니다. Verified Access 엔드포인트에 대한 정책 문서를 정의하지 않은 경우 Verified Access 그룹 정책에만 액세스가 필요합니다.

> **Note**
>
> AWS Verified Access 정책을 만들 때 구문의 유효성을 검사하지만 조건문에 입력한 데이터의 유효성은 확인하지 않습니다.

Verified Access 정책을 위한 기본 제공 연산자

에서 검증된 액세스 정책 설명 구조 설명한 대로 다양한 조건을 사용하여 AWS Verified Access 정책 컨텍스트를 만들 때 && 연산자를 사용하여 조건을 추가할 수 있습니다. 정책 조건에 추가적인 표현력을 추가하는 데 사용할 수 있는 다른 내장 연산자도 많이 있습니다. 다음 표에는 참조용으로 제공되는 모든 내장 연산자가 나와 있습니다.

연산자	유형 및 오버로드	설명
!	Boolean → Boolean	논리 not.
==	any → any	대등. 유형이 일치하지 않는 경우에도 모든 유형의 값에서 작동합니다. 서로 다른 유형의 값은 결코 서로 같을 수 없습니다.
!=	any → any	부등, 대등의 정반대(위 참조).
<	(long, long) → Boolean	보다 작은 배장 정수.
<=	(long, long) → Boolean	긴 정수 less-than-or-equal -to.
>	(long, long) → Boolean	보다 큰 배장 정수.
>=	(long, long) → Boolean	긴 정수 greater-than-or-equal -to.

연산자	유형 및 오버로드	설명
인	(entity, entity) → Boolean	계층 멤버십(재귀적: A의 A는 항상 참임).
	(entity, set(entity)) → Boolean	계층 멤버십: (A와 B) || (C의 A) ||이면 [B, C,...] 의 A는 참이며 ... 집합에 개체가 아닌 항목이 포함된 경우 오류입니다.
&&	(Boolean, Boolean) → Boolean	논리 및 (단락 평가).
||	(Boolean, Boolean) → Boolean	논리 또는 (단락 평가).
.exists()	entity → Boolean	엔터티 존재.
has	(entity, attribute) → Boolean	중위 연산자. `e has f`은(는) 레코드 또는 엔터티 `e`에 속성 `f`에 대한 바인딩이 있는지 테스트합니다. `e`가 존재하지 않는 경우 또는 `e`가 존재하지만 속성 `f`가 없는 경우 `false`를 반환합니다. 속성은 식별자 또는 문자열로 표현할 수 있습니다.
like	(string, string) → Boolean	중위 연산자. `t like p`은(는) 텍스트 `t`가 패턴 `p`과 일치하는지 확인합니다. 패턴에는 0개 이상의 문자와 일치하는 와일드카드 문자 `*`가 포함될 수 있습니다. `t`에서 문자 그대로의 별표 문자를 일치시키려면 `p`의 `\*`에서 특수 문자열을 사용할 수 있습니다.
.contains()	(set, any) → Boolean	멤버십을 설정합니다(B는 A의 요소인지).

연산자	유형 및 오버로드	설명
.containsAll()	(set, set) → Boolean	A 집합에 B 집합의 모든 요소가 포함되어 있는지 테스트합니다.
.containsAny()	(set, set) → Boolean	A 집합에 B 집합의 요소가 포함되어 있는지 테스트합니다.

검증된 액세스 정책 설명

AWS Verified Access 정책에 의견 설명을 포함할 수 있습니다. 의견은 //로 시작하고 새 줄로 끝나는 줄로 정의됩니다.

다음 예제에서는 정책의 의견 설명을 보여줍니다.

```
// 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)
};
```

검증된 액세스 정책 로직이 종료되었습니다.

특정 상황에 존재하거나 존재하지 않을 수 있는 데이터를 평가하는 AWS Verified Access 정책을 작성하는 것이 좋습니다. 존재하지 않는 컨텍스트의 데이터를 참조하는 경우 Cedar는 오류를 생성하고 사용자의 의도와 상관없이 정책을 평가하여 액세스를 거부합니다. 예를 들어, 이 컨텍스트에 `fake_provider`와(과) `bogus_key`이(가) 존재하지 않으므로 거부로 이어질 수 있습니다.

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

이러한 상황을 피하려면 has 연산자를 사용하여 키가 있는지 확인할 수 있습니다. has 연산자가 거짓을 반환하면 연결된 문장에 대한 추가 평가가 중단되고 Cedar는 존재하지 않는 항목을 참조하려고 시도하면서 오류를 발생하지 않습니다.

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

이는 서로 다른 두 신뢰 공급자를 참조하는 정책을 지정할 때 가장 유용합니다.

```
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)
    )
  )
};
```

검증된 액세스 예제 정책

예 1: IAM ID 센터를 위한 정책 생성

> **Note**
>
> 그룹 이름을 변경할 수 있으므로 IAM Identity Center는 그룹 ID를 사용하여 그룹을 참조합니다. 이렇게 하면 그룹 이름을 변경할 때 정책 설명이 위반되는 것을 방지할 수 있습니다.

다음 예제 정책은 사용자가 finance 그룹(c242c5b0-6081-1845-6fa8-6e0d9513c107의 그룹 ID를 가짐)에 속하고 확인된 이메일 주소를 가진 경우에만 액세스를 허용합니다.

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

예 1b: IAM ID 센터의 정책 설명에 조건 추가

다음 예제 정책은 사용자가 finance 그룹(c242c5b0-6081-1845-6fa8-6e0d9513c107의 그룹 ID를 가짐)에 속하고 확인된 이메일 주소를 갖고 있으며 Jamf 디바이스 위험 점수가 LOW인 경우에만 액세스를 허용합니다.

```
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"
};
```

예 2: 타사 OIDC 공급자에 대한 동일한 정책

다음 예제 정책은 사용자가 “금융” 그룹에 속하고 이메일 주소를 확인했으며 Jamf 기기 위험 점수가 다음과 같은 경우에만 액세스를 허용합니다. LOW

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

예 3: 사용 CrowdStrike

다음 예제 정책은 전체 평가 점수가 50점을 넘을 때 액세스를 허용합니다.

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

```
};
```

예제 4: 특수 문자로 작업

다음 예제는 컨텍스트 속성이 정책 언어의 예약 문자인 :(세미콜론)을 사용하는 경우 정책을 작성하는 방법을 보여줍니다.

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

예제 5: 특정 IP 주소 허용

다음 예제는 특정 IP 주소만 허용하는 정책을 보여줍니다.

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

예제 5a: 특정 IP 주소 차단

다음 예제는 특정 IP 주소를 차단하는 정책을 보여줍니다.

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

Verified Access 정책 도우미

Verified Access 정책 도우미는 정책을 테스트하고 개발하는 데 사용할 수 있는 검증된 액세스 콘솔의 도구입니다. 엔드포인트 정책, 그룹 정책 및 신뢰 컨텍스트를 한 화면에 표시하여 정책을 테스트하고 편집할 수 있습니다.

신뢰 컨텍스트 형식은 신뢰 공급자마다 다르며 Verified Access 관리자는 특정 신뢰 제공자가 사용하는 정확한 형식을 모를 수도 있습니다. 따라서 테스트 및 개발 목적으로 한 곳에서 신뢰 컨텍스트와 그룹 및 엔드포인트 정책을 모두 확인하는 것이 매우 유용할 수 있습니다.

다음 섹션에서는 환경의 기능에 대해 설명합니다.

Tasks

- 1단계: 리소스 지정
- 2단계: 정책 테스트 및 편집
- 3단계: 변경 사항 검토 및 적용

1단계: 리소스 지정

다음 섹션에서는, 사용하려는 AMI를 선택합니다. 또한 사용자(이메일 주소로 식별)를 지정하고 선택적으로 사용자 이름 및/또는 장치 식별자를 지정합니다. 기본적으로 가장 최근의 승인 결정은 지정된 사용자의 Verified Access 로그에서 추출됩니다. 선택적으로 가장 최근의 허용 또는 거부 결정을 구체적으로 선택할 수 있습니다.

마지막으로 신뢰 컨텍스트, 권한 부여 결정, 엔드포인트 정책 및 그룹 정책이 모두 다음 화면에 표시됩니다.

정책 도우미를 열고 리소스를 지정하는 방법

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified Access 인스턴스를 선택한 다음 사용할 인스턴스의 Verified Access 인스턴스 ID를 클릭합니다.
3. 정책 지원 시작을 선택합니다.
4. 사용자 이메일 주소에 사용자의 이메일 주소를 입력합니다.
5. Verified Access 엔드포인트에서 정책을 편집하고 테스트하려는 엔드포인트를 선택합니다.
6. (선택 사항) 이름에는 사용자 이름을 입력합니다.
7. (선택 사항) 장치 식별자에 고유한 장치 식별자를 입력합니다.
8. (선택 사항) 인증 결과에서 사용하려는 최근 인증 결과의 유형을 선택합니다. 기본적으로 최신 권한 부여 결과가 사용됩니다.
9. Next(다음)를 선택합니다.

2단계: 정책 테스트 및 편집

이 페이지에는 작업할 때 사용할 수 있는 다음과 같은 정보가 표시됩니다.

- 신뢰 제공자가 사용자에게 보낸 신뢰 컨텍스트 및 (선택적으로) 이전 단계에서 지정한 장치입니다.

- 이전 단계에서 지정된 Verified Access 엔드포인트에 대한 Cedar 정책입니다.
- 엔드포인트가 속한 Verified Access 그룹에 대한 Cedar 정책입니다.

Verified Access 엔드포인트 및 그룹에 대한 Cedar 정책은 이 페이지에서 편집할 수 있지만 신뢰 컨텍스트는 정적입니다. 이제 이 페이지에서 Cedar 정책과 함께 신뢰 컨텍스트를 볼 수 있습니다.

정책 테스트 버튼을 선택하여 신뢰 컨텍스트와 비교하여 정책을 테스트하면 권한 부여 결과가 화면에 표시됩니다. 정책을 편집하고 변경 사항을 다시 테스트하여 필요에 따라 프로세스를 반복할 수 있습니다.

정책 변경에 만족하면 다음을 선택하여 정책 지원의 다음 화면으로 계속 진행합니다.

3단계: 변경 사항 검토 및 적용

정책 도우미의 마지막 페이지에서 쉽게 검토할 수 있도록 강조 표시된 정책 변경 내용을 확인할 수 있습니다. 이제 마지막으로 검토하고 변경 사항 적용을 선택하여 변경 사항을 적용할 수 있습니다.

이전을 선택하여 이전 페이지로 돌아가거나 취소를 선택하여 정책 지원을 완전히 취소할 수도 있습니다.

Verified·Access의 보안

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처를 활용할 수 있습니다.

보안은 기업과 기업 간의 공동 책임입니다. AWS 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. AWS Verified Access에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램별 범위 내AWS 서비스 (규정 준수 프로그램별) 를 참조하십시오.
- 클라우드에서의 보안 — 사용하는 AWS 서비스에 따라 책임이 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Verified·Access 사용 시 책임 분담 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 Verified·Access를 구성하는 방법을 보여줍니다. 또한 Verified Access 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

내용

Verified·Access의 데이터 보호

AWS 공동 책임 모델 공동 책임 모델 AWS Verified Access의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시를 참조하십시오FAQ. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 공동 책임 모델 및AWS 보안 GDPR 블로그의 블로그 게시물을 참조하십시오.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 개별 사용자에게 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM) 를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 다단계 인증 (MFA) 을 사용하십시오.
- SSLTLS/를 사용하여 AWS 리소스와 통신하세요. TLS1.2가 필요하고 TLS 1.3을 권장합니다.
- API를 사용하여 사용자 활동 로깅을 설정합니다 AWS CloudTrail.
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API an을 AWS 통해 액세스할 때 FIPS 140-3개의 검증된 암호화 모듈이 필요한 경우 엔드포인트를 사용하십시오. FIPS 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준 () 140-3을 참조하십시오. FIPS

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔을 사용하거나 API AWS CLI, 또는 다른 AWS 서비스 방법을 사용하여 검증된 액세스 권한을 사용하는 경우가 포함됩니다. AWS SDKs 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL a를 제공하는 경우 해당 서버에 대한 요청을 URL 검증하기 위해 자격 증명 정보를 에 포함하지 않는 것이 좋습니다.

전송 중 암호화

검증된 액세스는 전송 계층 보안 (TLS) 1.2 이상을 사용하여 인터넷을 통해 최종 사용자로부터 검증된 액세스 엔드포인트로 전송되는 모든 데이터를 암호화합니다.

인터네트워크 트래픽 개인 정보 보호

Verified Access를 구성하여 내 특정 리소스에 대한 액세스를 제한할 수 있습니다. VPC 사용자 기반 인증의 경우 엔드포인트에 액세스하는 사용자 그룹을 기반으로 네트워크 일부에 대한 액세스를 제한할 수도 있습니다. 자세한 내용은 Verified·Access 정책 단원을 참조하십시오.

AWS 검증된 액세스를 위한 유휴 데이터 암호화

AWS Verified Access는 기본적으로 AWS 소유 KMS 키를 사용하여 저장된 데이터를 암호화합니다. 저장 데이터를 기본적으로 암호화하면 민감한 데이터를 보호하는 데 수반되는 운영 오버헤드와 복잡성

을 줄이는 데 도움이 됩니다. 동시에 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다. 다음 섹션에서는 Verified Access가 저장된 데이터 암호화에 KMS 키를 사용하는 방법에 대한 세부 정보를 제공합니다.

내용

검증된 액세스 및 KMS 키

AWS 소유 키

Verified Access는 KMS 키를 사용하여 개인 식별 정보를 자동으로 암호화합니다 ()PII. 이는 기본적으로 발생하며 AWS 소유한 키의 사용을 직접 확인, 관리, 사용 또는 감사할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 소유 키를 참조하십시오.

이 암호화 계층을 비활성화하거나 다른 암호화 유형을 선택할 수는 없지만 Verified Access 리소스를 생성할 때 고객 관리 키를 선택하여 기존 AWS 소유 암호화 키 위에 두 번째 암호화 계층을 추가할 수 있습니다.

고객 관리형 키

Verified·Access는 생성 및 관리하는 대칭 고객 관리형 키를 사용하여 기존 기본 암호화에 두 번째 암호화 계층을 추가할 수 있도록 지원합니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.

- 키 정책 수립 및 유지
- IAM정책 및 권한 수립 및 유지 관리
- 키 정책 활성화 및 비활성화
- 키 암호화 자료 교체
- 태그 추가

- 키 별칭 생성
- 삭제를 위한 스케줄 키

자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키를 참조하십시오.

> **Note**
>
> Verified Access는 AWS 소유 키를 사용하여 저장된 데이터를 자동으로 암호화하여 개인 식별 데이터를 무료로 보호합니다.
> 하지만 고객 관리 키를 사용할 때는 AWS KMS 요금이 부과됩니다. 요금에 대한 자세한 내용은 AWS Key Management Service 요금을 참조하십시오.

개인 식별 정보

다음 표에는 Verified Access에서 사용하는 개인 식별 정보 (PII) 와 암호화 방법이 요약되어 있습니다.

데이터 유형	AWS 소유 키 암호화	고객 관리형 키 암호화 (선택 사항)
`Trust provider (user-type)` 사용자 유형 신뢰 공급자에는 AuthorizationEndpoint, UserInfoEndpoint, ClientId ClientSecret, 등과 같은 OIDC 옵션이 PII 고려됩니다.	활성화됨	활성화됨
`Trust provider (device-type)` 장치 유형 신뢰 공급자에는 a가 TenantId 포함되며 이는 고려됩니다. PII	활성화됨	활성화됨
`Group policy`	활성화됨	활성화됨

데이터 유형	AWS 소유 키 암호화	고객 관리형 키 암호화 (선택 사항)
Verified·Access 그룹을 생성하거나 수정하는 동안 제공됩니다. 액세스 요청 승인에 대한 규칙을 포함합니다. 사용자 이름, 이메일 주소 PII 등을 포함할 수 있습니다.		
`Endpoint policy` Verified·Access 엔드포인트를 생성하거나 수정하는 동안 제공됩니다. 액세스 요청 승인에 대한 규칙을 포함합니다. 사용자 이름 및 이메일 주소 PII 등을 포함할 수 있습니다.	활성화됨	활성화됨

AWS Verified Access에서 권한 부여를 사용하는 방법 AWS KMS

Verified·Access는 고객 관리형 키를 사용할 수 있는 권한이 필요합니다.

고객 관리 키로 암호화된 Verified Access 리소스를 생성하면 Verified Access에서 CreateGrant요청을 보내 사용자를 대신하여 권한 부여를 생성합니다 AWS KMS. 권한 AWS KMS 부여는 Verified Access에 계정의 고객 관리 키에 대한 액세스 권한을 부여하는 데 사용됩니다.

Verified·Access는 다음 내부 작업에 대해 고객 관리형 키를 사용할 수 있는 권한이 필요합니다.

- 암호화된 데이터 키를 복호화하는 AWS KMS 데 사용할 수 있도록 암호화된 데이터 키의 암호를 해독해 달라는 암호 해독 요청을 보내십시오.
- 권한 삭제 RetireGrant요청을 보내세요. AWS KMS

언제든지 권한 부여에 대한 액세스 권한을 취소하거나 고객 관리형 키에 대한 서비스 액세스를 제거할 수 있습니다. 그렇게 하면 Verified·Access는 고객 관리형 키로 암호화된 데이터에 액세스할 수 없으며, 이는 해당 데이터에 의존하는 작업에 영향을 미칩니다.

Verified·Access로 고객 관리형 키 사용

또는 를 사용하여 대칭 고객 관리 키를 생성할 수 있습니다. AWS Management Console AWS KMS APIs AWS Key Management Service 개발자 안내서의 대칭 고객 관리형 키 생성 단계를 따르십시오.

키 정책

키 정책은 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 생성할 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키에 대한 액세스 관리를 참조하십시오.

Verified Access 리소스에서 고객 관리 키를 사용하려면 키 정책에서 다음 API 작업을 허용해야 합니다.

- `kms:CreateGrant` - 고객 관리형 키에 권한 부여를 추가합니다. 지정된 KMS 키에 대한 제어 액세스 권한을 부여하여 Verified Access에 필요한 작업을 허용하는 액세스 권한을 허용합니다. 권한 부여 사용에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하십시오.

  이를 통해 Verified·Access는 다음을 수행할 수 있습니다.
  - 데이터 키가 암호화에 즉시 사용되지 않으므로 암호화된 데이터 키를 생성하고 저장하려면 `GenerateDataKeyWithoutPlainText`를 호출합니다.
  - 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하려면 `Decrypt`를 호출합니다.
  - `RetireGrant`에 대한 서비스를 허용하도록 사용 중지 주체를 설정합니다.
- `kms:DescribeKey` – Verified·Access에서 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.
- `kms:GenerateDataKey` - Verified·Access가 키를 사용하여 데이터를 암호화할 수 있도록 허용합니다.
- `kms:Decrypt` - Verified·Access가 암호화된 데이터 키를 해독할 수 있도록 허용합니다.

다음은 Verified·Access에 사용할 수 있는 키 정책의 예입니다.

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

정책에서의 권한 지정에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하십시오.

키 액세스 문제 해결에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하십시오.

Verified·Access 리소스에 대한 고객 관리형 키 지정

고객 관리형 키를 지정하여 다음 리소스에 2차 계층 암호화를 제공할 수 있습니다.

- Verified·Access 그룹
- Verified·Access 엔드포인트
- Verified·Access 신뢰 공급자

를 사용하여 이러한 리소스를 생성하는 경우 추가 암호화 -- 선택 사항 섹션에서 고객 관리 키를 지정할 수 있습니다. AWS Management Console프로세스 중에 암호화 설정 사용자 지정 (고급) 확인란을 선택한 다음 사용하려는 AWS KMS 키 ID를 입력합니다. 기존 리소스를 수정하거나 AWS CLI를 사용하여 이 작업을 수행할 수도 있습니다.

> **Note**
>
> 위 리소스에 암호화를 추가하는 데 사용한 고객 관리 키가 손실되면 해당 리소스의 구성 값에 더 이상 액세스할 수 없습니다. 하지만 AWS Management Console 또는 AWS CLI를 사용하여 새 고객 관리 키를 적용하고 구성 값을 재설정하여 리소스를 수정할 수 있습니다.

AWS 검증된 액세스 암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보를 포함하는 선택적 키-값 쌍 집합입니다. AWS KMS 암호화 컨텍스트를 인증된 추가 데이터로 사용하여 인증된 암호화를 지원합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하면 암호화 컨텍스트를 암호화된 데이터에 AWS KMS 바인딩합니다. 요청에 동일한 암호화 컨텍스트를 포함해야 이 데이터를 해독할 수 있습니다.

AWS 검증된 Access 암호화 컨텍스트

Verified Access는 모든 AWS KMS 암호화 작업에서 동일한 암호화 컨텍스트를 사용합니다. 여기서 키는 `aws:verified-access:arn` 이고 값은 리소스 Amazon Resource Name (ARN) 입니다. 다음은 Verified·Access 리소스의 암호화 컨텍스트입니다.

Verified·Access 신뢰 공급자

```
"encryptionContext": {
    "aws:verified-access:arn":
```

```
    "arn:aws:ec2:region:111122223333:VerifiedAccessTrustProviderId"
}
```

Verified·Access 그룹

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

Verified·Access 엔드포인트

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

권한 부여 또는 정책에서의 암호화 컨텍스트 사용에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 암호화 컨텍스트를 참조하십시오.

AWS 검증된 액세스를 위한 암호화 키 모니터링

고객 관리 KMS 키를 Verified Access 리소스와 함께 사용하는 경우 AWS Verified Access가 보내는 요청을 추적하는 AWS CloudTrail데 사용할 수 AWS KMS있습니다.

다음은 고객 관리 KMS 키로 암호화된 데이터에 액세스하기 위해 `CreateGrant` Verified Access에서 호출한 KMS 작업을 모니터링하는`GenerateDataKey`,,, 및 에 대한 AWS CloudTrail 이벤트입니다. `RetireGrant` `Decrypt` `DescribeKey`

CreateGrant

고객 관리형 키를 사용하여 리소스를 암호화하는 경우 Verified·Access는 사용자를 대신하여 AWS 계정의 키에 액세스하라는 `CreateGrant` 요청을 보냅니다. Verified·Access에서 생성하는 권한 부여는 고객 관리형 키와 연결된 리소스에만 적용됩니다.

다음 예제 이벤트는 `CreateGrant` 작업을 기록합니다.

```
{
    "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·Access는 리소스를 삭제할 때 `RetireGrant` 작업을 사용하여 권한 부여를 제거합니다.

다음 예제 이벤트는 `RetireGrant` 작업을 기록합니다.

```
{
    "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·Access는 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하는 `Decrypt` 작업을 호출합니다.

다음 예제 이벤트는 Decrypt 작업을 기록합니다.


```
{
    "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·Access는 DescribeKey 작업을 사용하여 리소스와 연결된 고객 관리형 키가 계정 및 리전에 존재하는지 확인합니다.

다음 예제 이벤트는 DescribeKey 작업을 기록합니다.

```
{
    "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

다음 예제 이벤트는 GenerateDataKey 작업을 기록합니다.

```
{
    "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·Access의 ID 및 액세스 관리

AWS Identity and Access Management (IAM) 는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와줍니다. IAM관리자는 Verified Access 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 관리합니다. IAM추가 비용 없이 사용할 AWS 서비스 수 있습니다.

주제

고객

Verified Access에서 수행하는 작업에 따라 AWS Identity and Access Management (IAM) 사용 방법이 다릅니다.

서비스 사용자 - Verified·Access 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 Verified·Access 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Verified·Access의 기능에 액세스할 수 없는 경우 Verified·Access 자격 증명 및 액세스 문제 해결를 참조하십시오.

서비스 관리자 - 회사에서 Verified·Access 리소스를 책임지고 있는 경우 Verified·Access에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Verified·Access 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 서비스 사용자의 권한을 변경해 달라는 요청을 제출해야 합니다. 이 페이지의 정보를 검토하여 의 기본 개념을 IAM 이해하십시오. 회사에서 Verified IAM Access를 사용하는 방법에 대한 자세한 내용은 을 참조하십시오검증된 액세스의 작동 방식 IAM.

IAM관리자 - IAM 관리자인 경우 Verified Access에 대한 액세스를 관리하기 위한 정책을 작성하는 방법에 대해 자세히 알아보는 것이 좋습니다. 에서 IAM 사용할 수 있는 Verified Access ID 기반 정책의 예를 보려면 을 참조하십시오. Verified·Access에 대한 자격 증명 기반 정책 예제

ID를 통한 인증

인증은 ID 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM사용자로서 또는 역할을 수임하여 인증 (로그인 AWS) 을 받아야 합니다. AWS 계정 루트 사용자 IAM

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAMID 센터) 사용자, 회사의 SSO (Single Sign-On) 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인하는 경우 관리자는 이전에 역할을 사용하여 ID 페더레이션을 설정했습니다. IAM 페더레이션을 AWS 사용하여 액세스하는 경우 간접적으로 역할을 수임하는 것입니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호로 서명할 수 있는 소프트웨어 개발 키트 (SDKCLI) 와 명령줄 인터페이스 () 가 AWS 제공됩니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 사용 IAM설명서의 AWS API요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, 계정 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 AWS 것을 권장합니다. 자세한 내용은 사용 설명서의 다단계 인증 및 사용 AWS IAM Identity Center 설명서의 다단계 인증 사용 (MFA) 을 IAM 참조하십시오. AWS

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 작업에 루트 사용자를 사용하지 않

을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 작업의 전체 목록은 사용 설명서의 루트 사용자 자격 증명이 필요한 작업을 참조하십시오. IAM

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS Directory Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAMIdentity Center에서 사용자 및 그룹을 만들거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 AWS 계정 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. ID 센터에 대한 자세한 내용은 IAM ID 센터란 IAM 무엇입니까? 를 참조하십시오. AWS IAM Identity Center 사용 설명서에서

IAM 사용자 및 그룹

IAM사용자란 단일 사용자 또는 애플리케이션에 대한 특정 권한을 가진 사용자 내의 ID입니다. AWS 계정 가능하면 암호 및 액세스 키와 같은 장기 자격 증명을 가진 IAM 사용자를 만드는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 특정 사용 사례에서 IAM 사용자의 장기 자격 증명이 필요한 경우에는 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 사용 설명서의 장기 자격 증명이 필요한 사용 사례에 대한 정기적인 액세스 키 IAM 교체를 참조하십시오.

IAM그룹은 IAM 사용자 컬렉션을 지정하는 ID입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 이름을 지정한 IAMAdmins그룹을 만들고 해당 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세히 알아보려면 사용 설명서의 역할 대신 IAM 사용자를 만드는 시기를 참조하십시오. IAM

IAM역할

IAM역할은 특정 권한을 AWS 계정 가진 사용자 내의 ID입니다. IAM사용자와 비슷하지만 특정인과 관련이 있는 것은 아닙니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을

맡을 수 있습니다. AWS CLI or AWS API 작업을 호출하거나 사용자 지정을 사용하여 역할을 수임할 수 URL 있습니다. 역할 사용 방법에 대한 자세한 내용은 사용 IAM설명서의 IAM역할 사용을 참조하십시오.

IAM임시 자격 증명이 있는 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션을 위한 역할에 대한 자세한 내용은 IAM사용 설명서의 타사 ID 제공자를 위한 역할 생성을 참조하십시오. IAMIdentity Center를 사용하는 경우 권한 집합을 구성합니다. ID가 인증된 후 액세스할 수 있는 대상을 제어하기 위해 IAM Identity Center는 권한 집합을 역할의 상관 관계와 연결합니다. IAM 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할이 역할을 맡아 특정 작업에 대해 일시적으로 다른 권한을 부여받을 수 있습니다. IAM
- 계정 간 액세스 - IAM 역할을 사용하여 다른 계정의 사용자 (신뢰할 수 있는 사용자) 가 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 하지만 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책 간의 차이점을 알아보려면 사용 설명서의 교차 계정 리소스 액세스를 참조하십시오. IAM IAM
- 서비스 간 액세스 — 일부는 다른 기능을 AWS 서비스 사용합니다. AWS 서비스예를 들어, 서비스를 호출하면 해당 서비스가 Amazon에서 애플리케이션을 EC2 실행하거나 Amazon S3에 객체를 저장하는 것이 일반적입니다. 서비스는직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS요청 시 적용되는 정책 세부 정보는 전달 액세스 세션을 참조하십시오.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 간주하는 IAM 역할입니다. IAM관리자는 내부에서 IAM 서비스 역할을 만들고, 수정하고, 삭제할 수 있습니다. 자세한 내용은 사용 설명서의 역할 만들기를 참조하여 권한을 위임하십시오IAM. AWS 서비스
  - 서비스 연결 역할 - 서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자

에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.

- Amazon에서 실행 중인 애플리케이션 EC2 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS API 요청을 보내는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS CLI EC2인스턴스 내에 액세스 키를 저장하는 것보다 이 방법이 더 좋습니다. EC2인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 만들어야 합니다. 인스턴스 프로필에는 역할이 포함되며, 이를 통해 EC2 인스턴스에서 실행 중인 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 내용은 사용 설명서의 IAM역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 IAM 참조하십시오.

IAM역할을 사용할지 IAM 사용자를 사용할지 알아보려면 사용 설명서의 IAM 역할 생성 시기 (사용자 대신) 를 IAM참조하십시오.

정책을 사용한 액세스 관리

정책을 만들고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON정책 문서의 구조 및 내용에 대한 자세한 내용은 IAM사용 설명서의 JSON 정책 개요를 참조하십시오.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. IAM관리자는 IAM 정책을 생성하여 필요한 리소스에서 작업을 수행할 수 있는 권한을 사용자에게 부여할 수 있습니다. 그러면 관리자가 역할에 IAM 정책을 추가할 수 있으며, 사용자는 역할을 수임할 수 있습니다.

IAM정책은 작업을 수행하는 데 사용하는 방법에 관계없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 에서 역할 정보를 가져올 수 AWS API 있습니다.

보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 만드는 방법을 알아보려면 사용 설명서의 IAM정책 생성을 참조하십시오. IAM

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책과 인라인 정책 중 하나를 선택하는 방법을 알아보려면 IAM사용 설명서의 관리형 정책과 인라인 정책 중 선택을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 IAM 정책에서는 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록 () ACLs

액세스 제어 목록 (ACLs) 은 리소스에 액세스할 수 있는 권한을 가진 주체 (계정 구성원, 사용자 또는 역할) 를 제어합니다. ACLs정책 문서 형식을 사용하지는 않지만 리소스 기반 정책과 JSON 비슷합니다.

지원하는 서비스의 VPC 예로는 Amazon S3와 Amazon이 ACLs 있습니다. AWS WAF자세한 내용은 Amazon 심플 스토리지 서비스 개발자 안내서의 액세스 제어 목록 (ACL) 개요를 참조하십시오. ACLs

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 ID 기반 정책이 IAM 엔티티 (IAM사용자 또는 역할) 에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 객체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 사용 IAM설명서의 IAM 엔티티의 권한 경계를 참조하십시오.
- 서비스 제어 정책 (SCPs) - SCPs 조직 또는 OU (조직 구성 단위) 에 대한 최대 권한을 지정하는 JSON AWS Organizations정책입니다. AWS Organizations 기업이 소유한 여러 AWS 계정 개를 그

룹화하고 중앙에서 관리하는 서비스입니다. 조직의 모든 기능을 사용하도록 설정하면 일부 또는 모든 계정에 서비스 제어 정책 (SCPs) 을 적용할 수 있습니다. 각 항목을 포함하여 구성원 계정의 엔티티에 대한 권한을 SCP AWS 계정 루트 사용자제한합니다. Organizations 및 SCPs 에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하십시오.

- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM사용 설명서의 세션 정책을 참조하십시오.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련된 경우 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM사용 설명서의 정책 평가 로직을 참조하십시오.

검증된 액세스의 작동 방식 IAM

를 사용하여 IAM 인증된 액세스에 대한 액세스를 관리하기 전에 검증된 액세스와 함께 사용할 수 있는 IAM 기능에 대해 알아보십시오.

IAM기능:	Verified·Access 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키	예
ACLs	아니요
ABAC(정책의 태그)	부분
임시 보안 인증	예

IAM기능:	Verified·Access 지원
보안 주체 권한	예
서비스 역할	아니요
서비스 링크 역할	예

Verified Access 및 기타 AWS 서비스가 대부분의 IAM 기능과 어떻게 작동하는지 자세히 알아보려면 IAM사용 IAM 설명서에서 함께 작동하는AWS 서비스를 참조하십시오.

Verified·Access에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 만드는 방법을 알아보려면 사용 설명서의 IAM정책 생성을 참조하십시오. IAM

IAMID 기반 정책을 사용하면 허용 또는 거부된 작업 및 리소스는 물론 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 사용 IAM설명서의 IAM JSON 정책 요소 참조를 참조하십시오.

Verified·Access에 대한 자격 증명 기반 정책 예제

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

계정 간 액세스를 활성화하려면 다른 계정의 전체 계정 또는 IAM 엔티티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 AWS 계정경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM사용 설명서의 계정 간 리소스 액세스를 참조하십시오. IAM

Verified·Access에 대한 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

정책 `Action` 요소는 JSON 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 작업이 없는 권한 전용 작업과 같은 몇 가지 예외가 있습니다. API 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

액세스 확인 작업 목록을 보려면 서비스 인증 EC2 참조의 Amazon이 정의한 작업을 참조하십시오.

Verified·Access의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
ec2
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

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

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access에 대한 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource`JSON정책 요소는 작업이 적용되는 하나 또는 여러 개의 객체를 지정합니다. 문장에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. Amazon 리소스 이름 (ARN) 을 사용하여 리소스를 지정하는 것이 가장 좋습니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

확인된 액세스 리소스 유형 및 해당 ARNs 유형의 목록을 보려면 서비스 인증 참조의 EC2 Amazon이 정의한 리소스를 참조하십시오. 각 리소스에 어떤 작업을 지정할 수 있는지 알아보려면 Amazon에서 정의한 작업을 참조하십시오EC2. ARN

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

Verified·Access에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 `OR` 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어 리소스에 IAM 사용자 이름이 태그가 지정된 경우에만 리소스에 대한 액세스 권한을 IAM 사용자에게 부여할 수 있습니다. 자세한 내용은 IAM사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하십시오.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM사용 설명서의AWS 글로벌 조건 컨텍스트 키를 참조하십시오.

확인된 액세스 조건 키 목록을 보려면 서비스 권한 부여 EC2 참조의 Amazon용 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업 및 리소스를 알아보려면 Amazon에서 정의한 작업을 참조하십시오 EC2.

Verified·Access 자격 증명 기반 정책의 예제를 보려면 Verified·Access에 대한 자격 증명 기반 정책 예제를 참조하십시오.

ACLs검증된 액세스에서

지원ACLs: 아니요

액세스 제어 목록 (ACLs) 은 리소스에 액세스할 수 있는 권한을 가진 주체 (계정 구성원, 사용자 또는 역할) 를 제어합니다. ACLs정책 문서 형식을 사용하지는 않지만 리소스 기반 정책과 JSON 비슷합니다.

ABAC검증된 액세스 포함

지원 ABAC (정책의 태그): 부분

속성 기반 액세스 제어 (ABAC) 는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM엔티티 (사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. 의 ABAC 첫 번째 단계는 엔티티와 리소스에 태그를 지정하는 것입니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC빠르게 성장하는 환경에서 유용하며 정책 관리가 복잡해지는 상황에도 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

에 대한 자세한 내용은 What is ABAC? 를 참조하십시오. ABAC IAM사용 설명서에서 설정 ABAC 단계가 포함된 자습서를 보려면 사용 IAM설명서의 속성 기반 액세스 제어 사용 (ABAC) 을 참조하십시오.

Verified·Access에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

임시 자격 증명을 사용하여 로그인하면 작동하지 AWS 서비스 않는 것도 있습니다. 임시 자격 증명을 사용할 수 있는 AWS 서비스 방법을 비롯한 추가 정보는 IAM사용 IAM 설명서에서 AWS 서비스 해당 자격 증명을 사용할 수 있는 항목을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하는 경우 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 Single Sign-On (SSO) 링크를 AWS 사용하여 액세스하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM사용 설명서의 역할 전환 (콘솔) 을 참조하십시오.

AWS CLI 또는 를 사용하여 임시 자격 증명을 수동으로 생성할 수 AWS API 있습니다. 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 내용은 의 임시 보안 자격 증명을 참조하십시오. IAM

Verified·Access의 서비스 간 보안 주체 권한

순방향 액세스 세션 지원 (FAS): 예

에서 IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 사용자는 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS를 호출하는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. AWS 서비스 FAS요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS요청 시 적용되는 정책 세부 정보는 전달 액세스 세션을 참조하십시오.

Verified·Access를 위한 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 간주하는 IAM역할입니다. IAM관리자는 내부에서 IAM 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 사용 설명서의 역할 만들기를 참조하여 권한을 위임하십시오IAM. AWS 서비스

Verified·Access를 위한 서비스 연결 역할

서비스 링크 역할 지원: 예

서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.

Verified·Access 서비스 연결 역할을 생성 또는 관리하는 방법에 대한 자세한 내용은 Verified·Access에 대한 서비스 연결 역할 사용 섹션을 참조하십시오.

Verified·Access에 대한 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 Verified·Access 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 를 사용하여 작업을 수행할 수 없습니다 AWS API. IAM관리자는 IAM 정책을 생성하여 필요한 리소스에서 작업을 수행할 수 있는 권한을 사용자에게 부여할 수 있습니다. 그러면 관리자가 역할에 IAM 정책을 추가할 수 있으며, 사용자는 역할을 수임할 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 만드는 방법을 알아보려면 사용 IAM설명서에서 IAM 정책 생성을 참조하십시오.

각 리소스 유형의 형식을 비롯하여 Verified EC2 Access에서 정의한 작업 및 리소스 유형에 ARNs 대한 자세한 내용은 서비스 인증 참조의 Amazon용 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- Verified·Access 인스턴스 생성 정책
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Verified·Access 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하십시오. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 내용은 IAM사용 설명서의 AWS 관리형 정책 또는 작업 기능에 대한AWS 관리형 정책을 참조하십시오.

- 최소 권한 적용 — IAM 정책으로 권한을 설정하는 경우 작업 수행에 필요한 권한만 부여하십시오. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. 를 사용하여 권한을 IAM 적용하는 방법에 대한 자세한 내용은 사용 설명서의 정책 및 권한을 참조하십시오. IAM IAM
- IAM정책의 조건을 사용하여 액세스를 추가로 제한 - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, 를 사용하여 모든 요청을 전송하도록 지정하는 정책 조건을 작성할 수 SSL 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 내용은 IAM사용 설명서의 IAMJSON정책 요소: 조건을 참조하십시오.
- IAMAccess Analyzer를 사용하여 IAM 정책을 검증하여 안전하고 기능적인 권한을 보장합니다. IAM Access Analyzer는 새 정책과 기존 정책을 검증하여 정책이 IAM 정책 언어 (JSON) 및 IAM 모범 사례를 준수하는지 확인합니다. IAMAccess Analyzer는 안전하고 기능적인 정책을 작성하는 데 도움이 되는 100개 이상의 정책 검사와 실행 가능한 권장 사항을 제공합니다. 자세한 내용은 사용 설명서의 IAMAccess Analyzer 정책 검증을 참조하십시오. IAM
- 다단계 인증 필요 (MFA) - 사용자 또는 루트 IAM 사용자가 필요한 시나리오가 있는 경우 보안을 강화하려면 이 기능을 MFA 켜십시오. AWS 계정 API작업 호출 MFA 시기를 요구하려면 정책에 MFA 조건을 추가하세요. 자세한 내용은 IAM사용 설명서의 MFA -보호된 API 액세스 구성을 참조하십시오.

의 모범 사례에 IAM 대한 자세한 내용은 IAM사용 설명서의 보안 모범 사례를 참조하십시오. IAM

Verified·Access 인스턴스 생성 정책

Verified Access 인스턴스를 만들려면 IAM 보안 주체가 이 추가 설명을 정책에 추가해야 합니다IAM.

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

> **Note**
>
> `verified-access:AllowVerifiedAccess`작업 전용 가상입니다. API 리소스, 태그 또는 조건 키 기반 권한 부여는 지원하지 않습니다. 작업에 리소스, 태그 또는 조건 키 기반 인증을 사용하십시오. `ec2:CreateVerifiedAccessInstance` API

Verified·Access 인스턴스 생성에 대한 예제 정책입니다. 이 예제에서 123456789012 는 AWS 계정 번호이고 us-east-1 는 지역입니다. AWS

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

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제에서는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 만드는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 OR를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 AWS CLI 권한이 포함됩니다. 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 자격 증명 및 액세스 문제 해결

다음 정보를 사용하면 Verified Access와 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 해결하는 데 도움이 IAM 됩니다.

문제

- Verified·Access에서 작업을 수행할 권한이 없음
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 외부 사용자가 내 Verified Access AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

Verified·Access에서 작업을 수행할 권한이 없음

작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 권한이 없는 경우 발생합니다. ec2:*GetWidget*

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

이 경우 ec2:*GetWidget*작업을 사용하여 *my-example-widget*리소스에 액세스할 수 있도록 mateojackson사용자 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Verified·Access에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 라는 IAM 사용자가 Verified Access에서 콘솔을 사용하여 작업을 `marymajor` 수행하려고 할 때 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

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

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 Verified Access AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록 (ACLs) 을 지원하는 서비스의 경우 이러한 정책을 사용하여 사용자에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- Verified·Access에서 이러한 기능을 지원하는지 여부를 알아보려면 검증된 액세스의 작동 방식 IAM를 참조하십시오.
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 사용 설명서에서 소유한 다른 IAM AWS 계정 사용자의 액세스 권한 제공을 IAM 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM사용 설명서의 제3자가 AWS 계정 소유한 리소스에 대한 액세스 제공을 참조하십시오. AWS 계정

- ID 페더레이션을 통해 액세스를 제공하는 방법을 알아보려면 사용 설명서의 외부 인증된 사용자에게 액세스 제공 (ID 페더레이션) 을 IAM 참조하십시오.
- 계정 간 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 사용 설명서의 계정 간 리소스 액세스를 참조하십시오. IAM IAM

Verified·Access에 대한 서비스 연결 역할 사용

AWS Verified Access AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 검증된 액세스에 직접 연결되는 고유한 IAM 역할 유형입니다. 서비스 연결 역할은 Verified Access에서 미리 정의되며 서비스가 사용자를 대신하여 다른 사람에게 전화를 거는 데 필요한 모든 권한을 포함합니다. AWS 서비스

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할은 Verified·Access를 더 쉽게 설정할 수 있습니다. Verified·Access에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한, Verified·Access만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 엔티티에 연결할 수 없습니다. IAM

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 함께 작동하는AWS 서비스를 IAM 참조하고 서비스 연결 역할 열에서 '예'로 표시된 서비스를 찾아보세요. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

Verified·Access에 대한 서비스 연결 역할 권한

Verified Access는 이름이 지정된 서비스 연결 역할을 사용하여 서비스를 사용하는 AWSServiceRoleForVPCVerifiedAccess데 필요한 리소스를 계정에 프로비저닝합니다.

AWSServiceRoleForVPCVerifiedAccess서비스 연결 역할은 다음 서비스를 신뢰하여 역할을 수임합니다.

- `verified-access.amazonaws.com`

이름이 지정된 AWSVPCVerifiedAccessServiceRolePolicy역할 권한 정책을 사용하면 Verified Access가 지정된 리소스에서 다음 작업을 완료할 수 있습니다.

- 모든 서브넷 및 보안 그룹과 `VerifiedAccessManaged=true` 태그가 있는 모든 네트워크 인터페이스에 대한 조치 `ec2:CreateNetworkInterface`
- 생성 시 모든 네트워크 인터페이스에 대한 조치 `ec2:CreateTags`

- `VerifiedAccessManaged=true` 태그가 있는 모든 네트워크 인터페이스에 대한 조치 `ec2:DeleteNetworkInterface`
- 모든 보안 그룹 및 `VerifiedAccessManaged=true` 태그가 있는 모든 네트워크 인터페이스에 대한 조치 `ec2:ModifyNetworkInterfaceAttribute`

에서 이 정책에 대한 권한을 보거나 AWS 관리형 정책 참조 가이드에서 AWSVPCVerifiedAccessServiceRolePolicy정책을 볼 수도 있습니다. AWS Management Console AWSVPCVerifiedAccessServiceRolePolicy

IAM엔티티 (예: 사용자, 그룹 또는 역할) 가 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 권한을 구성해야 합니다. 자세한 내용은 사용 설명서의 서비스 연결 역할 권한을 참조하십시오. IAM

Verified·Access에 대한 서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 를 호출하면 Verified CreateVerifiedAccessEndpointAccess에서 서비스 연결 역할을 자동으로 생성합니다. AWS API

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. CreateVerifiedAccessEndpoint다시 전화를 걸면 Verified Access에서 서비스 연결 역할을 다시 생성합니다.

Verified·Access에 대한 서비스 연결 역할 편집

검증된 액세스를 사용하면 AWSServiceRoleForVPCVerifiedAccess서비스 연결 역할을 편집할 수 없습니다. 서비스 링크 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 를 사용하여 역할에 대한 설명을 편집할 수 있습니다. IAM 자세한 내용은 사용 IAM설명서의 서비스 연결 역할 편집을 참조하십시오.

Verified·Access에 대한 서비스 연결 역할 삭제

역할을 수동으로 삭제할 필요는 없습니다. AWSServiceRoleForVPCVerifiedAccess AWS Management Console AWS CLI, 또는 를 DeleteVerifiedAccessEndpoint호출하면 Verified Access가 리소스를 정리하고 서비스 연결 역할을 자동으로 삭제합니다. AWS API

를 사용하여 서비스 연결 역할을 수동으로 삭제하려면 IAM

IAM콘솔 AWS CLI, 또는 를 AWS API 사용하여 AWSServiceRoleForVPCVerifiedAccess서비스 연결 역할을 삭제합니다. 자세한 내용은 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오. IAM

Verified·Access 서비스 연결 역할이 지원되는 리전

Verified Access는 서비스를 사용할 수 AWS 리전 있는 모든 지역에서 서비스 연결 역할을 사용할 수 있도록 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 섹션을 참조하십시오.

AWS 검증된 액세스를 위한 관리형 정책

AWS 관리형 정책은 에서 생성하고 관리하는 독립 실행형 정책입니다. AWS AWS 관리형 정책은 많은 일반 사용 사례에 대한 권한을 제공하도록 설계되었으므로 사용자, 그룹 및 역할에 권한을 할당하기 시작할 수 있습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한 권한을 부여하지 않을 수도 있다는 점에 유의하세요. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

관리형 정책에 정의된 권한은 변경할 수 없습니다. AWS AWS 관리형 정책에 정의된 권한을 업데이트하는 경우 AWS 해당 업데이트는 정책이 연결된 모든 주체 ID (사용자, 그룹, 역할) 에 영향을 미칩니다. AWS 새 정책이 시작되거나 기존 서비스에 새 AWS 서비스 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 가장 높습니다.

자세한 내용은 IAM사용 설명서의 AWS 관리형 정책을 참조하십시오.

AWS 관리형 정책: AWSVPCVerifiedAccessServiceRolePolicy

이 정책은 Verified·Access가 사용자를 대신하여 작업을 수행할 수 있도록 서비스 연결 역할에 연결됩니다. 자세한 내용은 서비스 연결 역할 사용 단원을 참조하십시오. 이 정책에 대한 권한을 보려면 AWSVPCVerifiedAccessServiceRolePolicy에서 확인하거나 AWS 관리형 AWSVPCVerifiedAccessServiceRolePolicy정책 참조 가이드에서 정책을 볼 수 있습니다. AWS Management Console

AWS 관리형 정책에 대한 검증된 액세스 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후 Verified Access에 대한 AWS 관리형 정책 업데이트에 대한 세부 정보를 확인하십시오. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 Verified Access 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AWSVPCVerifiedAccessServiceRolePolicy- 정책 업데이트	Verified Access는 “sid” 필드 아래에 모든 작업에 대한 설명을 포함하도록 관리형 정책을 업데이트했습니다.	2023년 11월 17일
AWSVPCVerifiedAccessServiceRolePolicy- 정책 업데이트	Verified Access는 보안 그룹 리소스를 `ec2:CreateNetworkInterface` 권한에 추가하도록 관리형 정책을 업데이트했습니다.	2023년 5월 31일
AWSVPCVerifiedAccessServiceRolePolicy- 새 정책	Verified·Access는 서비스를 사용하는 데 필요한 리소스를 계정에 프로비저닝할 수 있도록 새로운 정책을 추가했습니다.	2022년 11월 29일
Verified·Access, 변경 내용 추적 시작	Verified Access는 AWS 관리형 정책의 변경 사항을 추적하기 시작했습니다.	2022년 11월 29일

Verified·Access의 규정 준수 확인

AWS Verified Access 연방 정보 처리 표준 (FIPS) 준수를 지원하도록 구성할 수 있습니다. Verified Access의 FIPS 규정 준수 설정에 대한 자세한 내용 및 자세한 내용은 을 참조하십시오FIPS검증된 액세스에 대한 규정 준수.

특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면AWS 서비스 규정 준수 프로그램의AWS 서비스 범위별, 규정 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 AWS 규정 준수 프로그램AWS 보증 프로그램 규정AWS 참조하십시오.

를 사용하여 AWS Artifact타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 퀵 스타트 가이드 - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 — 이 백서에서는 기업이 적합한 애플리케이션을 만드는 AWS HIPAA 데 사용할 수 있는 방법을 설명합니다.

> **Note**
>
> 모든 AWS 서비스 사람이 자격이 있는 것은 아닙니다. HIPAA 자세한 내용은 HIPAA적격 서비스 참조를 참조하십시오.

- AWS 규정AWS 준수 리소스 — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (국립 표준 기술 연구소 (NIST), 결제 카드 산업 보안 표준 위원회 (), 국제 표준화 기구 ()) 를 포함한PCI) 전반의 보안 제어에 대한 지침을 매핑합니다. ISO
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub— 이를 AWS 서비스 통해 내부 AWS보안 상태를 포괄적으로 파악할 수 있습니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty — 환경에 의심스럽고 악의적인 활동이 있는지 AWS 계정모니터링하여 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하는 PCI DSS 등 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 수 있습니다.
- AWS Audit Manager— 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

Verified·Access의 복원성

AWS 글로벌 인프라는 가용 영역을 중심으로 AWS 리전 구축됩니다. AWS 리전 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다. 이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가

이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

가용 영역에 대한 AWS 리전 자세한 내용은 글로벌 인프라를 참조하십시오AWS .

AWS 글로벌 인프라 외에도 Verified Access는 고가용성 요구 사항을 지원하는 데 도움이 되는 다음과 같은 기능을 제공합니다.

고가용성을 위한 다중 서브넷

로드 밸런서 유형의 Verified·Access 엔드포인트를 생성할 때 엔드포인트에 서브넷을 여러 개 연결할 수 있습니다. 엔드포인트와 연결하는 각 서브넷은 서로 다른 가용 영역에 속해야 합니다. 서브넷을 여러 개 연결하면 여러 가용 영역을 사용하여 고가용성을 보장할 수 있습니다.

모니터링 AWS Verified Access

모니터링은 의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다 AWS Verified Access. AWS Verified Access를 감시하고, 문제 발생 시 보고하고, 적절한 경우 자동 조치를 취할 수 있는 다음과 같은 모니터링 도구를 제공합니다.

- 액세스 로그 - 애플리케이션 액세스 요청에 대한 세부 정보를 캡처합니다. 자세한 내용은 the section called “Verified·Access 로그” 단원을 참조하십시오.
- AWS CloudTrail— 사용자 또는 사용자를 대신하여 이루어진 API 호출 AWS 계정 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 어떤 사용자와 계정이 전화를 걸었는지 AWS, 어떤 소스 IP 주소에서 호출이 이루어졌는지, 언제 호출이 발생했는지 식별할 수 있습니다. 자세한 내용은 the section called “CloudTrail 로그” 단원을 참조하십시오.

Verified·Access 로그

각 액세스 요청을 AWS Verified Access 평가한 후 모든 액세스 시도를 기록합니다. 이를 통해 애플리케이션 액세스에 대한 중앙 집중식 가시성을 확보하고 보안 사고 및 감사 요청에 신속하게 대응할 수 있습니다. Verified Access는 개방형 사이버 보안 스키마 프레임워크 (OCSF) 로깅 형식을 지원합니다.

로깅을 활성화할 때는 로그를 전송할 대상을 구성해야 합니다. 로깅 대상을 구성하는 데 사용되는 IAM 보안 주체에 로깅이 제대로 작동하려면 특정 권한이 있어야 합니다. 각 로깅 대상에 필요한 IAM 권한은 검증된 액세스 로깅 권한 섹션에서 확인할 수 있습니다. Verified·Access는 액세스 로그를 게시하기 위한 다음 대상을 지원합니다.

- 아마존 CloudWatch 로그 로그 그룹
- Amazon S3 버킷
- Amazon Data Firehose 전송 스트림

내용

- OCSF검증된 액세스에 대한 버전 1.0.0-rc.2 로그 예제

검증된 액세스 로깅 버전

기본적으로 검증된 액세스 로깅 시스템은 개방형 사이버 보안 스키마 프레임워크 (OCSF) 버전 0.1을 사용합니다. 버전 0.1을 사용하는 샘플 로그는 섹션 OCSF검증된 액세스에 대한 버전 0.1 로그 예제에서 확인할 수 있습니다.

최신 로깅 버전은 OCSF 버전 1.0.0-rc.2와 호환됩니다. 스키마에 대한 구체적인 세부 정보는 스키마에서 확인할 수 있습니다. OCSF 버전 1.0.0-rc.2를 사용하는 샘플 로그는 섹션 OCSF검증된 액세스에 대한 버전 1.0.0-rc.2 로그 예제에서 확인할 수 있습니다.

사용 중인 로깅 버전을 업그레이드하려면 다음 절차를 사용하십시오.

콘솔을 사용하여 로깅 버전을 업그레이드하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 적절한 Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 업데이트 로그 버전 드롭다운 목록에서 ocsf-1.0.0-rc.2를 선택합니다.
6. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 로깅 버전을 업그레이드하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용합니다.

검증된 액세스 로깅 권한

로깅 대상을 구성하는 데 사용되는 IAM 보안 주체에 로깅이 제대로 작동하려면 특정 권한이 있어야 합니다. 다음 섹션에서는 각 로깅 대상에 필요한 권한을 보여줍니다.

CloudWatch Logs로 전송하는 경우:

- Verified·Access 인스턴스에서 `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`

- 모든 리소스에서 `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDelivery`, `logs:ListLogDeliveries`, 및 `logs:UpdateLogDelivery`
- 대상 로그 그룹에서 `logs:DescribeLogGroups`, `logs:DescribeResourcePolicies`, 및 `logs:PutResourcePolicy`

Amazon S3로 전송하려면:

- Verified·Access 인스턴스에서 `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`
- 모든 리소스에서 `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDelivery`, `logs:ListLogDeliveries`, 및 `logs:UpdateLogDelivery`
- 그리고 대상 버킷에서 `s3:GetBucketPolicy` 및 `s3:PutBucketPolicy`

Firehose로 배송하는 경우:

- Verified·Access 인스턴스에서 `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`
- 모든 리소스에서 `firehose:TagDeliveryStream`
- 모든 리소스에서 `iam:CreateServiceLinkedRole`
- 모든 리소스에서 `logs:CreateLogDelivery`, `logs:DeleteLogDelivery`, `logs:GetLogDelivery`, `logs:ListLogDeliveries`, 및 `logs:UpdateLogDelivery`

검증된 액세스 로그 활성화 또는 비활성화

이 섹션의 절차를 사용하여 로깅을 활성화하거나 비활성화할 수 있습니다. 로깅을 활성화할 때는 로그를 전송할 대상을 구성해야 합니다. 로깅 대상을 구성하는 데 사용되는 IAM 보안 주체에 로깅이 제대로 작동하려면 특정 권한이 있어야 합니다. 각 로깅 대상에 필요한 IAM 권한은 검증된 액세스 로깅 권한 섹션에서 확인할 수 있습니다.

내용

액세스 로그 활성화

Verified·Access 로그를 활성화하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. (선택 사항) 신뢰 공급자가 보낸 신뢰 데이터를 로그에 포함하려면 다음을 수행하십시오.
   a. 업데이트 로그 버전 드롭다운 목록에서 ocsf-1.0.0-rc.2를 선택합니다.
   b. 신뢰 컨텍스트 포함을 선택합니다.
6. 다음 중 하나를 수행하십시오.
   - Amazon CloudWatch Logs로 전송을 켜십시오. 대상 로그 그룹을 선택합니다.
   - Amazon S3로 전송을 활성화합니다. 대상 버킷의 이름, 소유자 및 접두사를 입력합니다.
   - Firehose에 전달을 켜십시오. 대상 전송 스트림을 선택합니다.
7. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 검증된 액세스 로그를 활성화하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용하십시오.

액세스 로그 비활성화

Verified·Access 인스턴스의 액세스 로그는 언제든지 비활성화할 수 있습니다. 액세스 로그를 비활성화하면 로그 데이터는 사용자가 삭제할 때까지 로그 대상에 남아 있습니다.

Verified·Access 로그를 비활성화하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 로그 전송을 끕니다.

6. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 검증된 액세스 로그를 비활성화하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용하십시오.

검증된 액세스 신뢰 컨텍스트를 활성화 또는 비활성화합니다.

신뢰 제공자가 보낸 신뢰 컨텍스트를 선택적으로 Verified Access 로그에 포함하도록 활성화할 수 있습니다. 이는 애플리케이션에 대한 액세스를 허용하거나 거부하는 정책을 정의할 때 유용할 수 있습니다. 활성화하면 `data` 필드 아래의 로그에서 신뢰 컨텍스트를 찾을 수 있습니다. 신뢰 컨텍스트가 비활성화된 경우 `data` 필드는 로 설정됩니다`null`. 로그에 신뢰 컨텍스트를 포함하도록 검증된 액세스를 구성하려면 다음 절차를 수행하십시오.

> **Note**
>
> Verified·Access 로그에 신뢰 컨텍스트를 포함하려면 최신 로깅 버전 `ocsf-1.0.0-rc.2`로 업그레이드해야 합니다. 다음 절차에서는 이미 로깅을 활성화했다고 가정합니다. 그렇지 않은 경우 전체 절차 액세스 로그 활성화를 참조하십시오.

내용

- 신뢰 컨텍스트 활성화
- 신뢰 컨텍스트 비활성화

신뢰 컨텍스트 활성화

콘솔을 사용하여 Verified·Access 로그에 신뢰 컨텍스트를 포함하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 적절한 Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 업데이트 로그 버전 드롭다운 목록에서 ocsf-1.0.0-rc.2를 선택합니다.
6. 신뢰 컨텍스트 포함을 활성화합니다.

7. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

다음을 사용하여 검증된 액세스 로그에 신뢰 컨텍스트를 포함하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용하십시오.

신뢰 컨텍스트 비활성화

로그에 더 이상 신뢰 컨텍스트를 포함하지 않으려면 다음 절차를 수행하여 신뢰 컨텍스트를 제거할 수 있습니다.

콘솔을 사용하여 Verified·Access 로그에서 신뢰 컨텍스트를 제거하려면

1. 에서 Amazon VPC 콘솔을 엽니다 https://console.aws.amazon.com/vpc/.
2. 탐색 창에서 Verified·Access 인스턴스를 선택합니다.
3. 적절한 Verified·Access 인스턴스를 선택합니다.
4. Verified·Access 인스턴스 로깅 구성 탭에서 Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.
5. 신뢰 컨텍스트 포함을 끕니다.
6. Verified·Access 인스턴스 로깅 구성 수정을 선택합니다.

를 사용하여 검증된 액세스 로그에서 신뢰 컨텍스트를 제거하려면 AWS CLI

modify-verified-access-instance-logging-configuration 명령을 사용하십시오.

OCSF검증된 액세스에 대한 버전 0.1 로그 예제

다음은 기본 로깅 OCSF 버전 0.1을 사용하는 샘플 로그입니다.

예시

- 다음과 같이 액세스 권한이 부여되었습니다. OIDC
- OIDC및 를 사용하여 액세스 권한이 부여되었습니다. JAMF
- OIDC및 를 사용하여 액세스 권한이 부여되었습니다. CrowdStrike
- 쿠키 누락으로 인한 액세스 거부
- 정책으로 인한 액세스 거부
- 알 수 없는 로그 항목

다음과 같이 액세스 권한이 부여되었습니다. OIDC

이 예제 로그 항목에서 Verified Access는 OIDC 사용자 신뢰 공급자를 통해 엔드포인트에 대한 액세스를 허용합니다.


```
{
    "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
```

```
}
```

OIDC및 를 사용하여 액세스 권한이 부여되었습니다. JAMF

이 예제 로그 항목에서 Verified Access는 JAMF 기기 신뢰 제공자 모두를 OIDC 통해 엔드포인트에 대한 액세스를 허용합니다.

```
{
    "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
}
```


OIDC및 를 사용하여 액세스 권한이 부여되었습니다. CrowdStrike

이 예제 로그 항목에서 Verified Access는 CrowdStrike 기기 신뢰 제공자 모두를 OIDC 통해 엔드포인트에 대한 액세스를 허용합니다.


```
{
    "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
}
```


쿠키 누락으로 인한 액세스 거부

이 예제 로그 항목에서 Verified·Access는 인증 쿠키가 누락되어 액세스를 거부합니다.


```
{
    "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
}
```


정책으로 인한 액세스 거부

이 예제 로그 항목에서 Verified·Access는 액세스 정책에 의해 요청이 허용되지 않기 때문에 인증된 요청을 거부합니다.


```
{
```

```
    "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
}
```


알 수 없는 로그 항목

이 예제 로그 항목에서 Verified·Access는 전체 로그 항목을 생성할 수 없으므로 알 수 없는 로그 항목을 내보냅니다. 이렇게 하면 모든 요청이 액세스 로그에 표시됩니다.


```
{
    "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
}
```


OCSF검증된 액세스에 대한 버전 1.0.0-rc.2 로그 예제

다음은 로깅 버전 1.0.0-rc.2를 사용하는 샘플 로그입니다. OCSF

내용

신뢰 컨텍스트가 포함된 액세스 권한 부여


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

```
}
```

신뢰 컨텍스트가 생략된 액세스 권한 부여

```
{
    "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
```

```
}
```

를 사용하여 검증된 액세스 API 통화를 기록합니다. AWS CloudTrail

AWS Verified Access는 사용자 AWS CloudTrail, 역할 또는 Verified AWS 서비스 Access에서 수행한 작업의 기록을 제공하는 서비스인 서비스와 통합됩니다. CloudTrail 검증된 액세스에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 Verified Access 콘솔에서의 통화와 Verified Access API 작업에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 검증된 액세스를 위한 CloudTrail 이벤트를 포함하여 Amazon S3 버킷에 이벤트를 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 Verified Access에 대한 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

자세한 CloudTrail 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

인증된 액세스 정보는 의 CloudTrail

CloudTrail 계정을 만들 AWS 계정 때 활성화됩니다. Verified Access에서 활동이 발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 이벤트 기록으로 CloudTrail 이벤트 보기를 참조하십시오.

인증된 액세스에 대한 이벤트를 AWS 계정포함하여 내 이벤트의 진행 중인 기록을 보려면 트레일을 생성하십시오. 트레일을 사용하면 CloudTrail Amazon S3 버킷에 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 AWS 서비스 취하도록 기타를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 다음에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신

모든 액세스 검증 작업은 Amazon Reference에 의해 CloudTrail 기록되며 Amazon EC2 API Reference에 문서화되어 있습니다. 예를 들어`CreateVerifiedAccessInstance`, 에 대한 호출

DeleteVerifiedAccessInstance 및 ModifyVerifiedAccessInstance 작업은 CloudTrail 로그 파일에 항목을 생성합니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 사용자 자격 증명으로 이루어졌는지 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하십시오.

Verified·Access 로그 파일 항목 이해

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일은 하나 이상의 로그 항목을 포함합니다. 이벤트는 모든 소스로부터 단일 요청을 나타냅니다. 여기에는 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보가 포함됩니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 CreateVerifiedAccessInstance 작업에 대한 CloudTrail 로그 항목을 보여줍니다.


```
{
    "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"
}
```

에 대한 할당량 AWS Verified Access

AWS 계정 Y에는 각각에 대해 기본 할당량 (이전에는 한도라고 함) 이 있습니다. AWS 서비스다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다.

AWS 계정-레벨 할당량

AWS 계정 인증된 액세스와 관련된 할당량은 다음과 같습니다.

명칭	기본값	조정 가능	설명
Verified·Access 인스턴스	5	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 인스턴스의 최대 수입니다.
Verified·Access 그룹	10	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 그룹의 최대 수입니다.
Verified·Access 신뢰 공급자	15	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 신뢰 공급자의 최대 수입니다.
Verified·Access 엔드포인트	50	예	현재 리전에서 고객이 생성할 수 있는 Verified·Access 엔드포인트의 최대 수입니다.

HTTP헤더

HTTP헤더의 크기 제한은 다음과 같습니다.

명칭	기본값	조정 가능
요청 라인	16K	아니요
단일 헤더	16K	아니요
전체 응답 헤더	32K	아니요

명칭	기본값	조정 가능
전체 요청 헤더	64K	아니요

OIDC클레임 크기

OIDC클레임 크기 한도는 다음과 같습니다.

명칭	기본값	조정 가능
OIDC클레임 크기	11K	아니요

Verified·Access 사용 설명서에 대한 문서 이력

다음 표에서는 Verified·Access에 대한 문서 릴리스를 소개합니다.

변경 사항	설명	날짜
AWS 관리형 정책 업데이트	검증된 액세스에 대한 AWS 관리형 IAM 정책이 업데이트되었습니다.	2023년 11월 17일
유휴 시(저장된) 데이터 암호화	AWS Verified Access는 기본적으로 AWS 소유 KMS 키를 사용하여 저장된 데이터를 암호화합니다.	2023년 9월 28일
FIPS규정 준수 지원	FIPS규정 준수를 위해 검증된 액세스를 구성합니다.	2023년 9월 26일
향상된 로깅	로그에 신뢰 컨텍스트를 추가하는 로깅 기능 추가.	2023년 6월 19일
AWS 관리형 정책 업데이트	검증된 액세스에 대한 AWS 관리형 IAM 정책이 업데이트되었습니다.	2023년 5월 31일
GA 릴리스	Verified·Access 사용 설명서의 GA 릴리스. AWS WAF 통합을 포함합니다.	2023년 4월 27일
미리 보기 릴리스	Verified·Access 사용 안내서 미리 보기 릴리스	2022년 11월 29일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.