概念
ONTAP 9

NetApp
February 12, 2026

目次

概念

ONTAPにおけるOAuth 2.0認可サーバーとアクセストークン

許可サーバは、OAuth 2.0許可フレームワークの中心的なコンポーネントとして、いくつかの重要な機能を担っています。

OAuth 2.0許可サーバ

許可サーバは、主にアクセス トークンの作成と署名を行います。このアクセス トークンには、クライアント アプリケーションが保護されたリソースに選択的にアクセスするためのIDと許可情報が格納されます。許可サーバは通常、相互に隔離されています。また、スタンドアロンの専用サーバとして導入したり、IDおよびアクセス管理製品の一部として導入したりと、その導入形式はさまざまです。

> ⓘ 認可サーバーには、特にOAuth 2.0の機能がより大規模なIDおよびアクセス管理製品やソリューションに組み込まれている場合、異なる用語が使用されることがあります。例えば、*アイデンティティプロバイダー（IdP）*という用語は、*認可サーバー*と同義語として使用されることがよくあります。

管理

アクセス トークンの発行に加えて、許可サーバは、関連する管理サービスも提供します。これは通常、Web ユーザ インターフェイスを介して行われます。たとえば、次のようなことを定義したり管理したりできます。

- ユーザとユーザ認証
- スコープ
- テナントとRealmを通じた管理分離
- ポリシーの適用
- さまざまな外部サービスへの接続
- その他のIDプロトコル（SAMLなど）のサポート

ONTAPは、OAuth 2.0標準に準拠した許可サーバと互換性があります。

ONTAPへの定義

1台以上の許可サーバをONTAPに定義する必要があります。ONTAPは、各サーバとのセキュアな通信を通じてトークンを検証したり、その他の関連タスクを実行したりして、クライアント アプリケーションを支援します。

ONTAPの設定の主な側面を以下に示します。詳細については、"OAuth 2.0の導入シナリオ"も参照してください。

アクセス トークンの検証方法と検証場所

アクセス トークンの検証には、2つのオプションがあります。

- ローカル検証

ONTAPは、トークンを発行した許可サーバから提供された情報に基づいて、アクセス トークンをローカルで検証できます。許可サーバから取得した情報は、ONTAPによってキャッシュされ、定期的に更新されます。

- リモート イントロスペクション

  リモート イントロスペクションを使用して、許可サーバでトークンを検証することもできます。イントロスペクションは、許可された当事者がアクセス トークンについて許可サーバに問い合わせることを可能にするプロトコルです。イントロスペクションを使えば、ONTAPでアクセス トークンから特定のメタデータを抽出し、トークンを検証することができます。ONTAPは、パフォーマンス上の理由から一部のデータをキャッシュします。

ネットワークの位置

ONTAPは、ファイアウォールの内側にある可能性があります。この場合は、設定の際にプロキシを指定する必要があります。

許可サーバを定義する方法

CLI、System Manager、REST API などの管理インターフェイスを使用して、ONTAP に認証サーバーを定義できます。例えば、CLI ではコマンド `security oauth2 client create`を使用します。

> `security oauth2 client create`
> の詳細については、link:https://docs.netapp.com/us-en/ontap-cli/security-oauth2-client-create.html["ONTAPコマンド リファレンス"^]をご覧ください。

許可サーバの数

1つのONTAPクラスタに対して、最大8台の許可サーバを定義できます。発行者または発行者 / オーディエンスのクレームが一意である限り、同じ許可サーバを同じONTAPクラスタに複数回定義できます。たとえば、Keycloakで異なるRealmを使用する場合は、常にこれが該当します。

ONTAPでサポートされるOAuth 2.0の機能

OAuth 2.0のサポートは、最初にONTAP 9.14.1で利用可能になりましたが、その後のリリースで引き続き強化されています。ONTAPでサポートされているOAuth 2.0の機能を以下に説明します。

特定のONTAPリリースで導入された機能は、以降のリリースでも引き継がれます。

ONTAP 9.16.1

ONTAP 9.16.1では、OAuth 2.0の標準機能が拡張され、ネイティブのEntra IDグループ用のEntra ID固有の拡張機能が追加されています。これにより、アクセス トークンで名前の代わりにGUIDを使用できます。さらに、このリリースでは、アクセス トークンの「roles」フィールドを使用してネイティブ アイデンティティ プロバイダのロールをONTAPのロールにマッピングできる外部ロール マッピングのサポートが追加されています。

ONTAP 9.14.1

ONTAP 9.14.1以降では、次のOAuth 2.0の標準機能を通じて、以下を使用しているアプリケーションに対して許可サーバがサポートされます。

- "RFC6749：OAuth 2.0 認可フレームワーク"および "RFC 7519: JSON Web Token（JWT）"に記載されている「iss」、「aud」、「exp」などの標準フィールドを備えたOAuth 2.0。これには、アクセストークン内の「upn」、「appid」、「sub」、「username」、「preferred_username」などのフィールドを通じてユーザーを一意に識別する機能も含まれています。
- 「group」フィールドを使用したグループ名用のADFSベンダー固有の拡張機能。
- 「group」フィールドを使用したグループUUID用のAzureベンダー固有の拡張機能。
- OAuth 2.0のアクセス トークン スコープ内の自己完結型および指定ロールを使用して許可をサポートするONTAPの拡張機能。これには、「scope」フィールドと「scp」フィールド、およびスコープ内のグループ名が含まれます。

OAuth 2.0アクセス トークンの用途

許可サーバによって発行されたOAuth 2.0アクセス トークンは、ONTAPによって検証され、REST APIクライアント要求のロールベースのアクセス制御に使用されます。

アクセス トークンを取得する

アクセス トークンは、REST APIを使用するONTAPクラスタに定義されている許可サーバから取得する必要があります。トークンを取得するには、許可サーバと直接やり取りする必要があります。

> ONTAPがアクセス トークンを発行したり、クライアントからの要求を許可サーバにリダイレクトしたりすることはありません。

トークンを要求する方法は、次のようないくつかの要因によって異なります。

- 許可サーバとその設定オプション
- OAuth 2.0のグラント タイプ
- 要求の発行に使用するクライアントやソフトウェア ツール

グラント タイプ

grant とは、OAuth 2.0 アクセストークンをリクエストおよび取得するために用いられる、ネットワークフローのセットを含む明確に定義されたプロセスです。クライアント、環境、セキュリティ要件に応じて、複数の異なるタイプの grant を使用できます。一般的な grant タイプのリストを下の表に示します。

グラント タイプ	概要
クライアント クレデンシャル	一般的なグラント タイプで、クレデンシャル（IDや共有秘密鍵など）のみを使用します。クライアントがリソースのオーナーと密接な信頼関係にあることが想定されています。
パスワード	リソース オーナー パスワード クレデンシャルは、リソース オーナーがクライアントとの信頼関係を確立している場合に使用できるグラント タイプです。また、レガシーHTTPクライアントからOAuth 2.0に移行する場合にも役立ちます。
許可コード	機密クライアントに最適なグラント タイプで、リダイレクトベースのフローに基づいています。アクセス トークンとリフレッシュ トークンのどちらを取得するためにも使用できます。

JWTのコンテンツ

OAuth 2.0アクセス トークンは、JWT形式です。そのコンテンツは、設定に基づいて許可サーバによって作成されます。ただし、クライアント アプリケーションはトークンを解読できません。クライアントには、トークンを検査したり、そのコンテンツを認識したりする理由がありません。

各JWTアクセス トークンには、一連のクレームが格納されています。クレームには、許可サーバの管理定義に基づいて発行者と許可の特性が記述されています。次の表は、標準に登録されているクレームの一部をまとめたものです。すべての文字列で、大文字と小文字が区別されます。

クレーム	キーワード	概要
Issuer	iss	トークンを発行したプリンシパルを特定します。クレームの処理はアプリケーションにより異なります。
Subject	sub	トークンのサブジェクトまたはユーザです。クレーム名は、グローバルまたはローカルで一意のものに限定されます。
Audience	aud	目的とするトークンの受信者です。文字列の配列として記述されます。
Expiration	exp	トークンが期限切れになり、拒否されるまでの期間です。

詳細については、 "RFC 7519：JSON Web Tokens"を参照してください。

クライアント許可

ONTAPクライアント許可の概要とオプション

ONTAP OAuth 2.0の実装は、柔軟性と堅牢性を考慮した設計になっていて、ONTAP環境の保護に必要な機能を提供します。これには、同時に指定できない設定オプションがいくつかあります。許可の決定は、最終的にはOAuth 2.0のアクセス トークンに含まれている、またはアクセス トークンから導き出されたONTAP RESTロールに基づいて行われます。

OAuth 2.0の認証を設定する場合にのみ"ONTAP RESTロール"を使用できます。以前のONTAPの従来のロールはサポートされていません。

ONTAPは、設定に基づいて最も適切な単一の認証オプションを適用します。ONTAPがクライアントアクセスを決定する方法の詳細については、"ONTAPによるアクセスの制御方法"を参照してください。

OAuth 2.0の自己完結型スコープ

これらのスコープには、1つ以上のカスタムRESTロールが含まれており、それぞれがアクセストークン内の単一の文字列にカプセル化されています。これらはONTAPロール定義とは独立しています。スコープ文字列は認可サーバで設定する必要があります。詳細については、"自己完結型OAuth 2.0スコープ"を参照してください。

ローカルONTAP RESTロール

組み込みまたはカスタムの単一の名前付きRESTロールを使用できます。名前付きロールのスコープ構文は**ontap-role-**<URL-encoded-ONTAP-role-name>です。例えば、ONTAPロールが `admin`の場合、スコープ文字列は `ontap-role-admin`になります。

ユーザ

アプリケーション「http」へのアクセスが定義されたアクセス トークン内のユーザ名を使用できます。ユーザは、定義された認証方法に基づいて、password、domain（Active Directory）、nsswitch（LDAP）の順にテストされます。

グループ

許可にONTAPグループを使用するように許可サーバを設定できます。ローカルONTAPの定義を調べてもアクセスの可否を判定できない場合は、Active Directory（「domain」）グループかLDAP（「nsswitch」）グループが使用されます。グループ情報は、次の2つの方法のいずれかで指定できます。

- OAuth 2.0のスコープ文字列

  グループメンバーシップを持つユーザーが存在しないクライアント資格情報フローを使用した機密アプリケーションをサポートします。スコープ名は **ontap-group-**<URL-encoded-ONTAP-group-name> とする必要があります。例えば、グループが「development」の場合、スコープ文字列は「ontap-group-development」になります。

- 「グループ」のクレーム

  これは、リソース オーナー（パスワード グラント）フローを使用してADFSによって発行されるアクセス トークンが対象です。

詳細については、"ONTAP で OAuth 2.0 または SAML IdP グループを使用する"を参照してください。

ONTAPの自己完結型OAuth 2.0スコープ

自己完結型スコープは、アクセス トークンで伝送される文字列です。それぞれが完結したカスタム ロール定義であり、ONTAPがアクセスの可否を判定するために必要なものがすべて含まれています。スコープは、ONTAP内で定義されているRESTロールとは別の、独立したものです。

スコープ文字列のフォーマット

基本的に、スコープは連続した文字列で表され、コロンで区切られた6つの値で構成されます。ここでは、スコープ文字列で使用されるパラメータについて説明します。

ONTAPリテラル

スコープは、小文字のリテラル値 `ontap`で始まる必要があります。これにより、スコープがONTAPに固有であることが識別されます。

クラスタ

スコープが適用されるONTAPクラスタを定義します。指定できる値は、次のとおりです。

- クラスタUUID

  単一のクラスタを特定します。

- アスタリスク（*）

スコープをすべてのクラスタに適用することを意味します。

ONTAP CLIコマンド `cluster identity show`を使用して、クラスタのUUIDを表示できます。指定しない場合は、スコープがすべてのクラスタに適用されます。"ONTAPコマンド リファレンス"の `cluster identity show`の詳細をご覧ください。

ロール

自己完結型スコープに含まれるRESTロールの名前です。この値は、ONTAPで確認されたり、ONTAPに定義されている既存のRESTロールと照合されたりすることはありません。この名前は、ロギングに使用されます。

アクセス レベル

この値は、スコープ内でAPIエンドポイントを使用する場合にクライアント アプリケーションに適用されるアクセス レベルを表します。次の表に、設定できる6つの値をまとめておきます。

アクセス レベル	概要
なし	指定したエンドポイントへのアクセスをすべて拒否します。
readonly	GETを使用した読み取りアクセスのみを許可します。
read_create	読み取りアクセスと、POSTを使用した新しいリソース インスタンスの作成を許可します。
read_modify	読み取りアクセスと、PATCHを使用した既存のリソースの更新を許可します。
read_create_modify	削除以外のアクセスをすべて許可します。許可される処理は、GET（読み取り）、POST（作成）、PATCH（更新）です。
all	フル アクセスを許可します。

SVM

スコープが適用されるクラスター内のSVMの名前。すべてのSVMを指定するには、*（アスタリスク）を使用します。

この機能はONTAP 9.14.1では完全にはサポートされていません。SVMパラメータは無視し、プレースホルダとしてアスタリスクを使用できます。 "ONTAPリリース ノート"を確認して、今後のSVMサポートについてチェックしてください。

REST API URI

リソースまたは関連リソースセットへの完全パスまたは部分パス。文字列は `/api`で始まる必要があります。値を指定しない場合、スコープはONTAPクラスタのすべてのAPIエンドポイントに適用されます。

スコープの例

自己完結型スコープの例を、いくつか紹介します。

ontap:*:joes-role:read_create_modify:*:/api/cluster

このロールを割り当てられたユーザーに `/cluster`エンドポイントへの読み取り、作成、および変更アクセス権を付与します。

CLI管理ツール

自己完結型スコープの管理を容易にし、エラーの発生を抑えるために、ONTAP は `security oauth2 scope`入力パラメータに基づいてスコープ文字列を生成する CLI コマンドを提供しています。

このコマンド `security oauth2 scope`には、入力内容に基づいて2つの使用例があります：

- CLIパラメータからスコープ文字列を生成

  このバージョンのコマンドを使用すると、入力したパラメータに基づいてスコープ文字列を生成できます。

- スコープ文字列からCLIパラメータを生成

  このバージョンのコマンドを使用すると、入力したスコープ文字列に基づいてコマンド パラメータを生成できます。

例

次の例は、スコープ文字列を生成するものです。コマンド例に続いて、出力結果も掲載しています。定義は、すべてのクラスタに適用されます。

```
security oauth2 scope cli-to-scope -role joes-role -access readonly -api
/api/cluster
```

```
ontap:*:joes-role:readonly:*:/api/cluster
```

`security oauth2 scope`の詳細については、link:https://docs.netapp.com/us-en/ontap-cli/search.html?q=security+oauth2+scope["ONTAPコマンドリファレンス"^]をご覧ください。

ONTAP での OAuth 2.0 外部ロールマッピング

外部ロールは、ONTAPで使用するように設定されたアイデンティティ プロバイダで定義されます。ONTAP CLIを使用して、これらの外部ロールとONTAPロールのマッピング関係を作成および管理できます。

ONTAP REST APIを使用して外部ロールマッピング機能を設定することもできます。詳細については、 "ONTAP自動化ドキュメント"をご覧ください。

アクセス トークン内の外部ロール

以下は、2つの外部ロールを含むJSONアクセス トークンの一部です。

```
  ...
  "appidacr": "1",
  "family_name": "User",
  "name": "Test User 1",
  "oid": "4c2215c7-6d52-40a7-ce71-096fa41379ba",
  "roles": [
    "Global Administrator",
    "Application Administrator"
  ],
  "ver": "1.0",
  ...
```

構成

外部ロール マッピング機能は、ONTAPコマンドライン インターフェイスを使用して管理できます。

作成

`security login external-role-mapping create`コマンドを使用して、ロールマッピング設定を定義できます。このコマンドおよび関連オプションを実行するには、ONTAPの*admin*権限レベルが必要です。

パラメータ

グループ マッピングの作成に使用するパラメータを以下に示します。

パラメータ	概要
external-role	外部のアイデンティティ プロバイダで定義されているロールの名前。
provider	アイデンティティ プロバイダの名前。これはシステムの識別子である必要があります。
ontap-role	外部ロールがマッピングされている既存のONTAPロールを示します。

例

```
security login external-role-mapping create -external-role "Global
Administrator" -provider entra -ontap-role admin
```

`security login external-role-mapping create`の詳細については、link:https://docs.netapp.com/us-en/ontap-cli/security-login-external-role-mapping-create.html["ONTAPコマンド リファレンス"^]をご覧ください。

その他のCLI処理

このコマンドでは、次のような追加の処理がサポートされます。

- 表示
- 変更
- 削除

関連情報

- "ONTAPコマンド リファレンス"

ONTAPによるクライアント アクセスの制御方法

OAuth 2.0を適切に設計、導入するには、ONTAPがどのように許可設定を使用してクライアントのアクセス可否を判定しているのかを理解しておく必要があります。ここでは、アクセスを制御するために使用する主な手順をONTAPリリース別に紹介します。

ONTAP 9.15.1では、OAuth 2.0の重要な更新はありませんでした。9.15.1リリースを使用している場合は、ONTAP 9.14.1の説明を参照してください。

関連情報

- "ONTAPでサポートされるOAuth 2.0の機能"

ONTAP 9.16.1

ONTAP 9.16.1では、OAuth 2.0の標準のサポートが拡張され、ネイティブのEntra IDグループ用のMicrosoft Entra ID固有の拡張機能と外部のロール マッピングが追加されています。

ONTAP 9.16.1のクライアント アクセスの制御

ステップ**1**：自己完結型スコープ

アクセス トークンに自己完結型のスコープが含まれている場合、ONTAPは最初にそれらのスコープを調べます。自己完結型スコープがない場合は、手順2に進みます。

1つ以上の自己完結型スコープが存在する場合、ONTAPは明示的な*ALLOW*または*DENY*の決定が下されるまで各スコープを適用します。明示的な決定が下されると、処理は終了します。

ONTAPが明示的にアクセスの可否を判定できない場合は、手順2に進みます。

ステップ**2**：ローカルロールフラグを確認する

ONTAPはブールパラメータ `use-local-roles-if-present`を調べます。このフラグの値は、ONTAPに定義された各認証サーバーに対して個別に設定されます。

- 値が `true`の場合は、手順 3 に進みます。
- 値が `false`の場合、処理は終了し、アクセスは拒否されます。

ステップ**3**：名前付き**ONTAP REST**ロール

アクセストークンの `scope`または `scp`フィールド、あるいはクレームとして名前付きRESTロールが含まれている場合、ONTAPはそのロールを使用してアクセス決定を行います。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。

指定RESTロールがない場合、またはロールが見つからない場合は、手順4に進みます。

ステップ**4**：ユーザー

アクセス トークンからユーザ名が抽出され、アプリケーション「http」にアクセスできるユーザとの照合が試みられます。ユーザは、認証方法に基づいて次の順序で検証されます。

- password
- domain（Active Directory）
- nsswitch（LDAP）

一致するユーザーが見つかった場合、ONTAPはそのユーザーに定義されたロールを使用してアクセスを決定します。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。

一致するユーザがいない場合、またはアクセス トークンにユーザ名が含まれていない場合は、手順5に進みます。

ステップ**5**：グループ

1つ以上のグループが含まれている場合、フォーマットが検査されます。グループがUUIDで表現されている場合は、内部グループマッピングテーブルが検索されます。一致するグループと関連付けられたロールがある場合、ONTAPはグループに定義されているロールを使用してアクセスを決定します。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。詳細については、"ONTAP で OAuth 2.0 または SAML IdP グループを使用する"をご覧ください。

グループが名前で表現され、ドメインまたはnsswitch認証が設定されている場合、ONTAPはそれぞれActive DirectoryまたはLDAPグループとの照合を試みます。グループが一致する場合、ONTAPはグループに定義されているロールを使用してアクセス判定を行います。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。

一致するグループがない場合、またはアクセス トークンにグループが含まれていない場合、アクセスは拒否され、処理は終了します。

ONTAP 9.14.1

サポートされている初期のOAuth 2.0は、OAuth 2.0の標準機能に基づいて、ONTAP 9.14.1で導入されました。

ONTAP 9.14.1のクライアント アクセスの制御

ステップ**1**：自己完結型スコープ

アクセス トークンに自己完結型のスコープが含まれている場合、ONTAPは最初にそれらのスコープを調べます。自己完結型スコープがない場合は、手順2に進みます。

1つ以上の自己完結型スコープが存在する場合、ONTAPは明示的な*ALLOW*または*DENY*の決定が下されるまで各スコープを適用します。明示的な決定が下されると、処理は終了します。

ONTAPが明示的にアクセスの可否を判定できない場合は、手順2に進みます。

ステップ**2**：ローカルロールフラグを確認する

ONTAPはブールパラメータ `use-local-roles-if-present`を調べます。このフラグの値は、ONTAPに定義された各認証サーバーに対して個別に設定されます。

- 値が `true`の場合は、手順 3 に進みます。
- 値が `false`の場合、処理は終了し、アクセスは拒否されます。

ステップ**3**：名前付き**ONTAP REST**ロール

アクセストークンの `scope`または `scp`フィールドに名前付きREST roleが含まれている場合、ONTAPはそのロールを使用してアクセスを決定します。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。

指定RESTロールがない場合、またはロールが見つからない場合は、手順4に進みます。

ステップ**4**：ユーザー

アクセス トークンからユーザ名が抽出され、アプリケーション「http」にアクセスできるユーザとの照合が試みられます。ユーザは、認証方法に基づいて次の順序で検証されます。

- password
- domain（Active Directory）
- nsswitch（LDAP）

一致するユーザーが見つかった場合、ONTAPはそのユーザーに定義されたロールを使用してアクセスを決定します。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。

一致するユーザがいない場合、またはアクセス トークンにユーザ名が含まれていない場合は、手順5に進みます。

ステップ**5**：グループ

1つ以上のグループが含まれていて、domainまたはnsswitchの許可が設定されている場合、ONTAPはそれらのグループとそれぞれActive DirectoryまたはLDAPグループとの照合を試みます。

グループが一致する場合、ONTAPはグループに定義されたロールを使用してアクセスを決定します。その結果は常に*ALLOW*または*DENY*となり、処理は終了します。

一致するグループがない場合、またはアクセス トークンにグループが含まれていない場合、アクセスは拒否され、処理は終了します。

ONTAPを使用したOAuth 2.0導入シナリオ

ONTAPに許可サーバを定義する際には、いくつかの設定オプションが用意されています。これらのオプションに基づいて、複数ある導入シナリオのいずれかを使用して、環境に適した許可サーバを定義できます。

設定パラメータの概要

ONTAPに許可サーバを定義する際には、いくつかの設定パラメータを使用できます。これらのパラメータは、通常はどの管理インターフェイスでもサポートされています。

> 個々のパラメータまたはフィールドに使用される名前は、ONTAP管理インターフェイスによって異なる場合があります。管理インターフェイスの違いに対応するために、表内の各パラメータには単一の汎用名が使用されています。文脈に応じて、特定のインターフェイスで使用される正確な名前に読み替えてください。

パラメータ	概要
Name	ONTAPで認識される許可サーバの名前です。
Application	定義が適用されるONTAP内部アプリケーション。これは*http*である必要があります。
Issuer URI	トークンを発行するサイトまたは組織を特定するパスを含むFQDNです。
Provider JWKS URI	ONTAPがアクセス トークンの検証に使用するJSON Webキー セットを取得するパスとファイル名を含むFQDNです。
JWKS refresh interval	ONTAPがプロバイダのJWKS URIから証明書情報を更新する頻度を定めた時間間隔です。値は、ISO-8601形式で指定します。
Introspection endpoint	ONTAPがイントロスペクションを通じたリモート トークン検証に使用するパスを含むFQDNです。
Client ID	許可サーバで定義されているクライアントの名前です。この値を含める場合は、インターフェイスに基づいて関連付けられたクライアント シークレットも指定する必要があります。
Outgoing proxy	ONTAPがファイアウォールの内側にある場合に、許可サーバへのアクセスを提供するために指定します。URIはcurl形式にする必要があります。
Use local roles if present	指定RESTロールやローカル ユーザなど、ローカルのONTAP定義が使用されているかどうかを判定するブール値フラグです。
Remote user claim	ONTAPがローカルユーザを照合するために使用する代替名。アクセストークン内の `sub`フィールドを使用して、ローカルユーザ名と照合します。
Audience	このフィールドは、アクセス トークンを使用できるエンドポイントを定義します。

導入シナリオ

以下に、一般的な導入シナリオをいくつか示します。トークン検証がONTAPによってローカルで実行されるか、認可サーバによってリモートで実行されるかに基づいて整理されています。各シナリオには、必要な設定オプションのリストが含まれています。設定コマンドの例については、"ONTAPでのOAuth 2.0の導入"を参照してください。

認可サーバを定義したら、ONTAP管理インターフェイスからその設定を表示できます。たとえば、ONTAP CLIで `security oauth2 client show`コマンドを使用します。

ローカル検証

次の導入シナリオは、トークン検証がONTAPによってローカルで実行されるものです。

自己完結型スコープを使用（プロキシなし）

OAuth 2.0の自己完結型スコープのみを使用する、最もシンプルな導入シナリオです。ローカルのONTAP ID定義は使用しません。指定が必要なパラメータは次のとおりです。

- Name
- Application（http）
- Provider JWKS URI
- Issuer URI

また、許可サーバにスコープを追加する必要があります。

自己完結型スコープを使用（プロキシあり）

この導入シナリオでは、OAuth 2.0の自己完結型スコープを使用します。ローカルのONTAP ID定義は使用しません。ただし、許可サーバがファイアウォールの内側にあるため、プロキシを設定する必要があります。指定が必要なパラメータは次のとおりです。

- Name
- Application（http）
- Provider JWKS URI
- Outgoing proxy
- Issuer URI
- Audience

また、許可サーバにスコープを追加する必要があります。

ローカル ユーザのロールとデフォルト ユーザ名のマッピングを使用（プロキシあり）

このデプロイメントシナリオでは、デフォルトの名前マッピングを持つローカルユーザーロールを使用します。リモートユーザークレームはデフォルト値 `sub`を使用するため、アクセストークンのこのフィールドはローカルユーザー名との照合に使用されます。ユーザー名は40文字以下である必要があります。認可サーバーはファイアウォールの背後にあるため、プロキシも設定する必要があります。以下のパラメータを含める必要があります：

- Name
- Application（http）
- Provider JWKS URI
- 存在する場合はローカルロールを使用する(`true`
- Outgoing proxy
- Issuer

ローカル ユーザがONTAPに定義されていることを確認する必要があります。

ローカル ユーザのロールと代替ユーザ名マッピングを使用（プロキシあり）

この導入シナリオでは、ローカル ユーザのロールと、ローカルONTAPユーザの照合に使用される代替ユーザ名を使用します。許可サーバがファイアウォールの内側にあるため、プロキシを設定する必要があります。指定が必要なパラメータは次のとおりです。

- Name
- Application（http）
- Provider JWKS URI
- 存在する場合はローカルロールを使用する(`true`
- Remote user claim
- Outgoing proxy
- Issuer URI
- Audience

ローカル ユーザがONTAPに定義されていることを確認する必要があります。

リモート イントロスペクション

次の導入設定は、ONTAPがイントロスペクションを介してリモートでトークン検証を実行する場合のものです。

自己完結型スコープを使用（プロキシなし）

OAuth 2.0の自己完結型スコープを使用する、シンプルな導入シナリオです。ONTAP ID定義は使用しません。次のパラメータを含める必要があります。

- Name
- Application（http）
- Introspection endpoint
- Client ID
- Issuer URI

スコープのほかに、許可サーバでクライアントとクライアント シークレットを定義する必要があります。

関連情報

- "セキュリティ oauth2 クライアント表示"

OAuth 2.0 相互 TLS を使用した ONTAP クライアント認証

セキュリティ上のニーズに応じて、オプションで相互TLS（mTLS）を設定して強力なクライアント認証を導入できます。OAuth 2.0環境の一部としてONTAPでmTLSを使用すると、アクセス トークンが、その発行を受けたクライアントしか使用できなくなります。

相互TLSとOAuth 2.0

Transport Layer Security（TLS）は、2つのアプリケーション（通常はクライアント ブラウザとWebサーバ）の間にセキュアな通信チャネルを確立するために使用されます。相互TLSは、クライアント証明書を通じた強力なクライアント識別を実現することにより、これを拡張したものです。OAuth 2.0を導入したONTAPクラスタで使用すると、送信者限定アクセス トークンを作成および使用できるので、基本的なmTLS機能が拡張されます。

送信者制約アクセストークンは、元々発行されたクライアントのみが使用できます。この機能をサポートするために、新しい確認クレーム(`cnf`がトークンに挿入されます。このフィールドには、アクセストークンの要求時に使用されたクライアント証明書のダイジェストを保持するプロパティ `x5t#S256`が含まれます。この値は、トークン検証の一環としてONTAPによって検証されます。送信者制約のない認可サーバによって発行されたアクセストークンには、追加の確認クレームは含まれません。

認証サーバごとにmTLSを使用するようにONTAPを設定する必要があります。たとえば、CLIコマンド `security oauth2 client`には、以下の表に示す3つの値に基づいてmTLS処理を制御するパラメータ `use-mutual-tls`が含まれています。

それぞれの設定で、結果とONTAPによって実行されるアクションは、設定パラメータの値、アクセス トークンの内容、クライアント証明書によって異なります。表内のパラメータは、制限が最も緩いものから最も厳しいものの順に並んでいます。

パラメータ	概要
なし	OAuth 2.0の相互TLS認証が、許可サーバで完全に無効になります。ONTAPは、トークンに確認クレームが含まれている場合や、TLS接続でクライアント証明書が提供されている場合であっても、mTLSクライアント証明書認証を実行しません。
request	OAuth 2.0 相互 TLS 認証は、クライアントが送信者制約アクセストークンを提示した場合に強制されます。つまり、アクセストークン内に確認クレーム（プロパティ `x5t#S256`を含む）が存在する場合にのみ mTLS が強制されます。これがデフォルト設定です。
必須	許可サーバによって発行されたすべてのアクセス トークンについて、OAuth 2.0の相互TLS認証が実行されます。したがって、すべてのアクセス トークンが送信者限定である必要があります。アクセス トークンに確認クレームが存在しない場合や、無効なクライアント証明書がある場合、認証とREST API要求は失敗します。

導入フローの概要

ONTAP環境でmTLSとOAuth 2.0を使用する際の一般的な手順を以下に示します。詳細については、 "RFC 8705：OAuth 2.0 相互TLSクライアント認証と証明書バインドアクセストークン"を参照してください。

ステップ**1**：クライアント証明書を作成してインストールする

クライアントIDの確立は、クライアントの秘密鍵を知っていることの証明がベースになります。対応する公開鍵は、クライアントから提示された署名付きX.509証明書に配置されます。クライアント証明書の大まかな作成手順は、次のとおりです。

1. 公開鍵と秘密鍵のペアを生成する
2. 証明書署名要求を生成する

3. CSRファイルを既知のCAに送信する
4. CAが要求を検証して署名済み証明書を発行する

クライアント証明書は通常、ローカルのオペレーティング システムにインストールしたり、curlなどの一般的なユーティリティで直接使用したりできます。

ステップ**2**：**mTLS**を使用するように**ONTAP**を設定する

ONTAPでmTLSを使用するには、設定が必要です。この設定は認証サーバごとに個別に行います。例えば、CLIでは、コマンド `security oauth2 client`にオプションパラメータ `use-mutual-tls`を指定します。詳細については、"ONTAPでのOAuth 2.0の導入"を参照してください。

ステップ**3**：クライアントがアクセストークンを要求する

クライアントは、ONTAPに設定された許可サーバにアクセス トークンを要求する必要があります。クライアント アプリケーションは、手順1で作成してインストールした証明書でmTLSを使用する必要があります。

ステップ**4**：認可サーバーがアクセストークンを生成する

認可サーバーはクライアントリクエストを検証し、アクセストークンを生成します。この一環として、クライアント証明書のメッセージダイジェストを作成し、確認クレーム（フィールド `cnf`）としてトークンに含めます。

ステップ**5**：クライアントアプリケーションがアクセストークンを**ONTAP**に提示

クライアントアプリケーションはONTAPクラスタに対してREST API呼び出しを行い、アクセストークンを*ベアラートークン*として認可リクエストヘッダーに含めます。クライアントは、アクセストークンのリクエストに使用したのと同じ証明書を使用してmTLSを使用する必要があります。

ステップ **6**：**ONTAP** がクライアントとトークンを検証します。

ONTAPは、HTTPリクエスト内のアクセストークンと、mTLS処理の一部として使用されるクライアント証明書を受け取ります。ONTAPは、まずアクセストークン内の署名を検証します。設定に基づいて、ONTAPはクライアント証明書のメッセージダイジェストを生成し、トークン内の確認クレーム*cnf*と比較します。2つの値が一致する場合、ONTAPは、APIリクエストを発行しているクライアントが、アクセストークンが最初に発行されたクライアントと同じであることを確認したことになります。

関連情報

- "セキュリティ OAuth2 クライアント"

著作権に関する情報

商標に関する情報