API の外部ユーザーの設定

バージョン要件：外部 AAA を使用するには、Threat Defense バージョン 6.3(0) 以降、および Threat Defense REST API v2 以降を実行している必要があります。

外部 RADIUS AAA サーバーを使用して Threat Defense REST API へのユーザーアクセスを認証および認可するようデバイスを設定できます。組み込みのローカル **[管理（admin）]** ユーザーアカウントの代わり、またはこのユーザーアカウントに加えて RADIUS ユーザーアカウントを使用できます。

外部 AAA を使用すると、さまざまな承認レベルを持つアカウントを定義できます。これにより、デバイスに設定変更を加えるユーザーを制限しながら、サポートスタッフに読み取り専用アクセス権を提供することができます。

次の手順では、RADIUS アカウントをセットアップして、認証および認可に外部 AAA を使用するようデバイスを設定するエンドツーエンドのプロセスについて説明します。

始める前に

外部承認を使用する場合、次の操作要因に留意してください。

- デバイスが高可用性向けに設定されている場合、アクティブユニットで外部承認を設定します。その後、承認設定の展開ジョブを実行して、スタンバイデバイスへのユーザーアクセスを許可する必要があります。
- 新規ユーザーがシステムにアクセスするたびに、そのユーザーに対してユーザーリソースが作成されます。設定を展開して、そのユーザーオブジェクトを保存する必要があります。

  （Threat Defense 6.6 より前のバージョン）高可用性（HA）モードで稼働している場合、ユーザーがスタンバイユニットにログインする前に設定を展開する必要があります。管理者または読み取り/書き込みユーザーのみが展開ジョブを開始できるため、最初の読み取り専用ユーザーは、ユーザーオブジェクトを保存するための設定を別のユーザーに展開してもらう必要があります。

  Threat Defense 6.6 以降では、HA の制限が削除されます。外部ユーザーは、最初にアクティブユニットにログインせずに、スタンバイユニットにログインして設定を展開することができます。ユーザーオブジェクトはスタンバイユニットでは作成されませんが、有効

なユーザー名/パスワードが指定されていれば、ユーザーの特性はキャッシュされ、ユーザーにアクセス権が付与されます。

手順

ステップ 1 RADIUS ユーザーアカウントでの承認権限の定義 （2 ページ）。

ステップ 2 RADIUS サーバーの定義 （3 ページ）。

ステップ 3 RADIUS サーバー用 AAA サーバーグループの作成 （4 ページ）。

ステップ 4 HTTPS アクセスの認証ソースとしての AAA サーバーグループの確立 （7 ページ）。

ステップ 5 **[POST/運用/展開（POST/operational/deploy）]** を使用して、展開ジョブを開始します。

curl コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'

'https://ftd.example.com/api/fdm/[最新（latest）]/operational/deploy'
```

変更の展開の詳細については、設定変更の導入を参照してください。

ステップ 6 外部ユーザーアクセスの確認 （10 ページ）。

- RADIUS ユーザーアカウントでの承認権限の定義 （2 ページ）
- RADIUS サーバーの定義 （3 ページ）
- RADIUS サーバー用 AAA サーバーグループの作成 （4 ページ）
- HTTPS アクセスの認証ソースとしての AAA サーバーグループの確立 （7 ページ）
- 外部ユーザーアクセスの確認 （10 ページ）

RADIUS ユーザーアカウントでの承認権限の定義

外部 RADIUS サーバーからの Threat Defense REST API へのアクセスを提供できます。RADIUS 認証および認可を有効にすることにより、さまざまなレベルのアクセス権を付与でき、すべてのユーザがローカル**管理者**アカウントを使用してログインする必要がなくなります。

（注） これらの外部ユーザーは、Device Manager についても認証されます。

ロールベースのアクセス制御（RBAC）を提供するには、RADIUS サーバ上のユーザアカウントを更新して **cisco-av-pair** 属性を定義します。この属性はユーザーアカウントで正しく定義されている必要があります。正しく定義されていないと、ユーザーの REST API へのアクセスが拒否されます。cisco-av-pair 属性でサポートされる値は、次のとおりです。

- **fdm.userrole.authority.admin** はフル管理者アクセスを提供します。これらのユーザは、ローカル**管理者**ユーザが実行できるすべてのアクションを実行できます。

- **fdm.userrole.authority.rw** は読み取り/書き込みアクセスを提供します。これらのユーザは、読み取り専用ユーザが実行できるすべてのアクションを実行でき、設定を編集および展開することもできます。ただし、システムクリティカルなアクションだけは制限されます。これには、アップグレードのインストール、バックアップの作成と復元、監査ログの表示、および他のユーザーのログオフが含まれます。
- **fdm.userrole.authority.ro** は読み取り専用アクセスを提供します。これらのユーザは、ダッシュボードと設定を表示できますが、変更できません。ユーザが変更しようとすると、権限が不足していることを示すエラー メッセージが表示されます。

RADIUS サーバーの定義

適切な認証権限を定義するためのユーザーアカウントをRADIUSサーバーで設定すると、REST APIへのアクセスを認証および認可するためにサーバーが使用するデバイスを設定できます。

POST /object/radiusidentitysources リソースを使用して、定義する各 RADIUS サーバーのオブジェクトを作成します。

手順

ステップ 1 RADIUS サーバーの JSON オブジェクト本文を作成します。

このコールで使用する JSON オブジェクトの例を次に示します。

```
{
  "name": "aaa-server-1",
  "description": "RADIUS server for API access.",
  "host": "172.16.246.220",
  "timeout": 10,
  "serverAuthenticationPort": 1812,
  "serverSecretKey": "secret123",
  "type": "radiusidentitysource"
}
```

その属性は次のとおりです。

- name：オブジェクト名。この名前は、RADIUSサーバーに定義されている名前と一致している必要はありません。
- [Description]：（オプション。） オブジェクトの説明。
- host：RADIUS サーバーの IP アドレスまたは完全修飾ホスト名。
- timeout：（オプション）次のサーバーに要求を送信する前にサーバーからの応答を待機する時間の長さ（1 ～ 300 秒）です。この属性を含めない場合のデフォルトは 10 秒です。
- serverAuthenticationPort：（オプション）RADIUS認証および承認が実行されるポート。この属性を含めない場合のデフォルトは 1812 です。

- serverSecretKey：（オプション）Threat Defense デバイスと RADIUS サーバー間でデータを暗号化するために使用される共有秘密キー。キーは、大文字と小文字が区別される最大 64 文字の英数字文字列です。スペースは使用できません。キーは、英数字または下線で開始する必要があります。特殊文字 $ & - _ . + @ を使用できます。文字列は、RADIUS サーバーで設定された文字列と一致している必要があります。秘密キーを設定していない場合、接続は暗号化されません。

ステップ 2 オブジェクトをポストします。

たとえば、curl コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
 -d '{
    "name": "aaa-server-1",
   "description": "RADIUS server for API access.",
   "host": "172.16.246.220",
   "timeout": 10,
   "serverAuthenticationPort": 1812,
   "serverSecretKey": "secret123",
   "type": "radiusidentitysource"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/object/radiusidentitysources'
```

ステップ 3 応答を確認します。

取得する応答コードは200である必要があります。正常な応答本文は次のようになります。秘密鍵などの機密情報は、応答ではマスク処理されていることに注意してください。

```
{
  "version": "nfamb3cr2jlyi",
  "name": "aaa-server-1",
  "description": "RADIUS server for API access.",
  "host": "172.16.246.220",
  "timeout": 10,
  "serverAuthenticationPort": 1812,
  "serverSecretKey": "*********",
  "capabilities": [
    "AUTHENTICATION",
    "AUTHORIZATION"
  ],
  "id": "1b962e3b-6e56-11e8-bd65-379fa8aaaba1",
  "type": "radiusidentitysource",
  "links": {
    "self": "https://ftd.example.com/api/fdm/[最新（latest）]/object/
radiusidentitysources/1b962e3b-6e56-11e8-bd65-379fa8aaaba1"
  }
}
```

RADIUS サーバー用 AAA サーバーグループの作成

RADIUS サーバー オブジェクトを作成した後、POST /object/radiusidentitysourcegroups リソースを使用して AAA グループを作成し、radiusidentitysource オブジェクトを含めます。

最大 16 台の RADIUS サーバーを RADIUS AAA サーバーグループに追加できます。これらのサーバーは相互にバックアップになる必要があります。つまり、同じユーザーアカウントリストを持つ必要があります。

手順

ステップ 1 RADIUS サーバーグループの JSON オブジェクト本文を作成します。

このコールで使用する JSON オブジェクトの例を次に示します。

```
{
  "name": "radius-group",
  "maxFailedAttempts": 3,
  "deadTime": 10,
  "description": "AAA RADIUS server group.",
  "radiusIdentitySources": [
    {
      "id": "1b962e3b-6e56-11e8-bd65-379fa8aaaba1",
      "type": "radiusidentitysource",
      "version": "nfamb3cr2jlyi",
      "name": "aaa-server-1"
    }
  ],
  "type": "radiusidentitysourcegroup"
}
```

その属性は次のとおりです。

- name：オブジェクト名。メンバー RADIUS サーバーで定義されているものと一致している必要はありません。
- maxFailedAttempts：（オプション）失敗したサーバーは、すべてのサーバーが失敗した後にのみ再アクティブ化されます。デッドタイムは、最後のサーバーが失敗した後にすべてのサーバーを再アクティブ化するまで待機する時間の長さ（0 ～ 1440分）です。この属性が含まれていない場合、デフォルトは 10 分です。
- deadTime：（オプション）次のサーバーを試行する前に、グループ内の RADIUS サーバーに送信された要求の失敗数（応答がなかった要求の数）。1 ～ 5 を指定できます。デフォルトは 3 です。最大失敗試行回数を超えると、システムはそのサーバーを故障としてマークします。

  特定の機能について、ローカルデータベースを使用するフォールバック方式を設定していて、グループ内のすべてのサーバーが応答に失敗した場合、そのグループは非応答と見なされ、フォールバック方式が試行されます。サーバー グループはデッド タイムの間、非応答とマークされたままになるため、その期間内に追加の AAA 要求でサーバー グループへの接続は試行されず、フォールバック方式がすぐに使用されます。
- [Description]：（オプション。） オブジェクトの説明。
- radiusIdentitySources：グループに含める RADIUS サーバーを定義する各 radiusidentitysource オブジェクトを定義する項目のグループ。[ブラケット]内に項目を入れます。各オブジェクトの属性およびシンタックスは次のとおりです。個々のオブジェクトから、id、version、および name 属性の値を取得します。その情報は、オブジェクトを作成する際に応答本文

に含まれます。**GET/object/radiusidentitysources** コールから情報を取得することもできます。type は、 radiusidentitysource である必要があります。

```
{
  "id": "1b962e3b-6e56-11e8-bd65-379fa8aaaba1",
  "type": "radiusidentitysource",
  "version": "nfamb3cr2jlyi",
  "name": "aaa-server-1"
}
```

ステップ 2 オブジェクトをポストします。

たとえば、curl コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
 -d '{
   "name": "radius-group",
   "maxFailedAttempts": 3,
   "deadTime": 10,
   "description": "AAA RADIUS server group.",
   "radiusIdentitySources": [
     {
       "id": "1b962e3b-6e56-11e8-bd65-379fa8aaaba1",
       "type": "radiusidentitysource",
       "version": "nfamb3cr2jlyi",
       "name": "aaa-server-1"
     }
   ],
   "type": "radiusidentitysourcegroup"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/object/radiusidentitysourcegroups'
```

ステップ 3 応答を確認します。

取得する応答コードは 200 である必要があります。正常な応答本文は次のようになります。

```
{
  "version": "7r572novdiyy",
  "name": "radius-group",
  "maxFailedAttempts": 3,
  "deadTime": 10,
  "description": "AAA RADIUS server group.",
  "radiusIdentitySources": [
    {
      "version": "nfamb3cr2jlyi",
      "name": "aaa-server-1",
      "id": "1b962e3b-6e56-11e8-bd65-379fa8aaaba1",
      "type": "radiusidentitysource"
    }
  ],
  "activeDirectoryRealm": null,
  "id": "0a7996ae-6e5b-11e8-bd65-dbab801c44b9",
  "type": "radiusidentitysourcegroup",
  "links": {
    "self": "https://ftd.example.com/api/fdm/[最新（latest）]/object/
radiusidentitysourcegroups/0a7996ae-6e5b-11e8-bd65-dbab801c44b9"
```

```
  }
}
```

HTTPS アクセスの認証ソースとしての AAA サーバーグループの確立

PUT /devicesettings/default/aaasettings/{objId} リソースを使用して、ユーザー認証のアイデンティティ ソースである RADIUS AAA サーバー グループを特定します。

POST メソッドはありません。システム認証に必要なオブジェクトはすでに存在しています。最初に GET を実行して、関連 ID とバージョンの値を確認する必要があります。

手順

ステップ 1 GET /devicesettings/default/aaasettings を使用して、aaasettings オブジェクトの属性を確認します。

curl コマンドは次のようになります。

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/[最新（latest）]/devicesettings/default/aaasettings'
```

たとえば、応答本文は次のようになります。この例は、ローカル アイデンティティ ソースが HTTPS アクセス用に定義されているソースであることを示しています。これは、REST API とは関係がない SSH アクセスでも使用されます。

```
{
  "items": [
    {
      "version": "du52clrtmawlt",
      "name": "HTTPS",
      "identitySourceGroup": {
        "version": "cynutari5ffkl",
        "name": "LocalIdentitySource",
        "id": "e3e74c32-3c03-11e8-983b-95c21a1b6da9",
        "type": "localidentitysource"
      },
      "description": null,
      "protocolType": "HTTPS",
      "useLocal": "NOT_APPLICABLE",
      "id": "00000003-0000-0000-0000-000000000007",
      "type": "aaasetting",
      "links": {
        "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
devicesettings/default/aaasettings/00000003-0000-0000-0000-000000000007"
      }
    },
    {
      "version": "fgkhvu4kwucgv",
      "name": "SSH",
      "identitySourceGroup": {
        "version": "cynutari5ffkl",
```

```
        "name": "LocalIdentitySource",
        "id": "e3e74c32-3c03-11e8-983b-95c21a1b6da9",
        "type": "localidentitysource"
      },
      "description": null,
      "protocolType": "SSH",
      "useLocal": "NOT_APPLICABLE",
      "id": "00000003-0000-0000-0000-000000000008",
      "type": "aaasetting",
      "links": {
        "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
devicesettings/default/aaasettings/00000003-0000-0000-0000-000000000008"
      }
    }
  ],
  "paging": {
    "prev": [],
    "next": [],
    "limit": 10,
    "offset": 0,
    "count": 2,
    "pages": 0
  }
}
```


ステップ 2 （オプション）表示範囲を限定するには、GET /devicesettings/default/aaasettings/{objId} を使用して、HTTPS AAA 設定オブジェクトのコピーを取得します。

PUT コールでは HTTPS オブジェクトのみ更新されます。SSH オブジェクトを更新する必要はありません。

この例では、HTTPS オブジェクトの ID は 00000003-0000-0000-0000-000000000007 であるため、curl コマンドは次のようになります。

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/[最新（latest）]/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007'
```

応答本文は次のようになります。


```
{
  "version": "ha4653ootep7z",
  "name": "HTTPS",
  "identitySourceGroup": {
    "version": "cynutari5ffkl",
    "name": "LocalIdentitySource",
    "id": "e3e74c32-3c03-11e8-983b-95c21a1b6da9",
    "type": "localidentitysource"
  },
  "description": null,
  "protocolType": "HTTPS",
  "useLocal": "NOT_APPLICABLE",
  "id": "00000003-0000-0000-0000-000000000007",
  "type": "aaasetting",
  "links": {
    "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
devicesettings/default/aaasettings/00000003-0000-0000-0000-000000000007"
  }
```


ステップ 3 AAA 管理アクセス用の JSON オブジェクト本文を作成します。

このコールで使用する JSON オブジェクトの例を次に示します。

```
{
  "version": "ha4653ootep7z",
  "name": "HTTPS",
  "identitySourceGroup": {
    "id": "0a7996ae-6e5b-11e8-bd65-dbab801c44b9",
    "type": "radiusidentitysourcegroup",
    "version": "7r572novdiyy",
    "name": "radius-group"
  },
  "description": null,
  "protocolType": "HTTPS",
  "useLocal": "BEFORE",
  "id": "00000003-0000-0000-0000-000000000007",
  "type": "aaasetting"
}
```

その属性は次のとおりです。

- version：HTTPS オブジェクトのバージョン。GET コールの応答本文でこの値を検索します。
- name：オブジェクト名、HTTPS。GET コールの応答本文でこの値を検索します。
- identitySourceGroup：RADIUS サーバー グループを特定します。サーバー グループの作成（または GET /object/radiusidentitysourcegroups コール）時に応答本文から id、version、および name の値を取得します。type は、radiusidentitysource である必要があります。
- [Description]：（オプション。） オブジェクトの説明。
- protocolType：このソースが適用されるプロトコル、HTTPS。
- useLocal：ローカル管理者ユーザー アカウントを含む、ローカル アイデンティティ ソースの使用方法。次のいずれかのオプションを入力します。
  - **[前（Before）]**：最初にローカルソースに照らしてユーザー名とパスワードがチェックされます。
  - **[後（After）]**：外部ソースを利用できない場合、またはユーザーアカウントが外部ソース内で見つからない場合にのみローカルソースがチェックされます。
  - [使用しない（Never）]：（非推奨）ローカル ソースがまったく使用されないため、[管理者（admin）] ユーザーとしてログインできません。

    注意 **[使用しない（Never）]** を選択すると、**[管理（admin）]** アカウントを使用して Device Manager にログインする、または API を使用することができなくなります。RADIUS サーバが使用できなくなった場合または RADIUS サーバのアカウント設定が間違っている場合は、システムがロックされます。

- id：HTTPS オブジェクトの ID 値。GET コールの応答本文でこの値を検索します。
- type：オブジェクト タイプ、aaasetting。

ステップ 4 オブジェクトを PUT します。

たとえば、curl コマンドは次のようになります。URL の {objId} と JSON オブジェクト内の aaasettings オブジェクトの ID は同じである点に注意してください。

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json'
 -d '{
   "version": "ha4653ootep7z",
   "name": "HTTPS",
   "identitySourceGroup": {
     "id": "0a7996ae-6e5b-11e8-bd65-dbab801c44b9",
     "type": "radiusidentitysourcegroup",
     "version": "7r572novdiyy",
     "name": "radius-group"
   },
   "description": null,
   "protocolType": "HTTPS",
   "useLocal": "BEFORE",
   "id": "00000003-0000-0000-0000-000000000007",
   "type": "aaasetting"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007'
```

ステップ 5 応答を確認します。

取得する応答コードは 200 である必要があります。正常な応答本文は次のようになります。

```
{
  "version": "ehxcytq4iccb3",
  "name": "HTTPS",
  "identitySourceGroup": {
    "version": "7r572novdiyy",
    "name": "radius-group",
    "id": "0a7996ae-6e5b-11e8-bd65-dbab801c44b9",
    "type": "radiusidentitysourcegroup"
  },
  "description": null,
  "protocolType": "HTTPS",
  "useLocal": "BEFORE",
  "id": "00000003-0000-0000-0000-000000000007",
  "type": "aaasetting",
  "links": {
    "self": "https://ftd.example.com/api/fdm/[最新（latest）]/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007"
  }
}
```

外部ユーザーアクセスの確認

展開ジョブが完了したら、Device Manager と REST API の両方への外部ユーザーアクセスをテストできます。

手順

ステップ 1 有効な cisco-av-pair 属性を持つ外部ユーザー名を使用して、Device Manager にログインします。

ログインが成功すると、ページの右上にユーザー名と権限レベルが表示されます。

ステップ 2 外部ユーザーの REST API トークンを取得します。

ユーザーがトークンを取得できる場合、そのユーザーは、割り当てられた権限レベルに許可されているリソースとメソッドを使用できます。

a) シンプルなパスワード付与トークンの場合は JSON オブジェクト本文を作成します。

```
{
  "grant_type": "password",
  "username": "radiusreadwriteuser1",
  "password": "Readwrite123!"
}
```

b) **POST /fdm/token** を使用して、トークンを取得します。

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{
   "grant_type": "password",
   "username": "radiusreadwriteuser1",
   "password": "Readwrite123!"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/fdm/token'
```

c) 応答を評価して、トークンが付与されたことを確認します。

取得する応答コードは 200 である必要があります。トークンの取得は、システムがユーザーを認証できたことを意味します。

```
{
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1Mjg4MjM
3MTAsInN1YiI6InJhZGl1c3JlYWR3cml0ZXVzZXIxIiwianRpIjoiMjk5Zj
Q5YjYtNmU2NC0xMWU4LWJkNjUtNmY0ZmVmYjY1MzI5IiwibmJmIjoxNTI4O
DIzNzEwLCJleHAiOjE1Mjg4MjU1MTAsInJlZnJlc2hUb2tlbkV4cGlyZXNB
dCI6MTUyODgyNjExMDg4OSwidG9rZW5UeXBlIjoiSldUX0FjY2VzcyIsInV
zZXJVdWlkIjoiMjliMjBlNjctNmU2NC0xMWU4LWJkNjUtMzU4MmUwZjU5Yj
Q4IiwidXNlclJvbGUiOiJST0xFX1JFQURfV1JJVEUiLCJvcmlnaW4iOiJwY
XNzd29yZCJ9.dtKsl9IB4ds3RAktEeaSuQy_Zs2SrzLr976UtblBt28",
  "expires_in": 1800,
  "token_type": "Bearer",
  "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1Mjg4Mj
M3MTAsInN1YiI6InJhZGl1c3JlYWR3cml0ZXVzZXIxIiwianRpIjoiMjk5Z
jQ5YjYtNmU2NC0xMWU4LWJkNjUtNmY0ZmVmYjY1MzI5IiwibmJmIjoxNTI4
ODIzNzEwLCJleHAiOjE1Mjg4MjYxMTAsImFjY2Vzc1Rva2VuRXhwaXJlc0F
0IjoxNTI4ODI1NTEwODg5LCJyZWZyZXNoQ291bnQiOi0xLCJ0b2tlblR5cG
UiOiJKV1RfUmVmcmVzaCIsInVzZXJVdWlkIjoiMjliMjBlNjctNmU2NC0xM
WU4LWJkNjUtMzU4MmUwZjU5YjQ4IiwidXNlclJvbGUiOiJST0xFX1JFQURf
V1JJVEUiLCJvcmlnaW4iOiJwYXNzd29yZCJ9.Lc7MYmieNMMrjx7XoTiW-x
8Z8qFCnzfNM1apgbwLQvo",
  "refresh_expires_in": 2400
}
```

ステップ 3 **GET /object/users** を使用して、各ユーザーに作成されているユーザーオブジェクトを確認します。

Device Manager にログインする、またはアクセストークンを取得する新規ユーザーのユーザーオブジェクトは自動的に作成されます。展開ジョブを実行して、それらのユーザーオブジェクトを保存する必要があります。ハイ アベイラビリティ モードでは、ユーザーがスタンバイ装置にログインする前に展開ジョブを実行する必要があります。

たとえば、curl コマンドは次のようになります。

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/[最新（latest）]/object/users'
```

次の応答本文は、2 人の外部ユーザーがログインしたことを示しています。**userRole** には、2 人のユーザーアカウントに対して、RADIUS サーバーに設定されている cisco-av-pair から取得した権限が表示されています。この情報を使用して、RADIUS ユーザーアカウントを正しく設定していることを確認します。**admin** ユーザーはローカルに定義されたユーザーです。

```
{
  "items": [
    {
      "version": "h2vom4wckm2js",
      "name": "radiusadminuser1",
      "password": null,
      "newPassword": null,
      "userPreferences": {
        "preferredTimeZone": "(UTC+00:00) UTC",
        "colorTheme": "NORMAL_CISCO_IDENTITY",
        "type": "userpreferences"
      },
      "userRole": "ROLE_ADMIN",
      "identitySourceId": "0a7996ae-6e5b-11e8-bd65-dbab801c44b9",
      "userServiceTypes": [
        "MGMT"
      ],
      "id": "150d9754-6e63-11e8-bd65-ed9b20f62114",
      "type": "user",
      "links": {
        "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
object/users/150d9754-6e63-11e8-bd65-ed9b20f62114"
      }
    },
    {
      "version": "p4rgwcjr5colj",
      "name": "admin",
      "password": null,
      "newPassword": null,
      "userPreferences": {
        "preferredTimeZone": "(UTC-07:00) America/Los_Angeles",
        "colorTheme": "NORMAL_CISCO_IDENTITY",
        "type": "userpreferences"
      },
      "userRole": "ROLE_ADMIN",
      "identitySourceId": "e3e74c32-3c03-11e8-983b-95c21a1b6da9",
      "userServiceTypes": [
        "MGMT"
      ],
      "id": "5023d3ab-6dc5-11e8-b9ed-db6dba9bf94c",
      "type": "user",
```

```
      "links": {
        "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
object/users/5023d3ab-6dc5-11e8-b9ed-db6dba9bf94c"
      }
    },
    {
      "version": "ngx7a2dixngoq",
      "name": "radiusreadwriteuser1",
      "password": null,
      "newPassword": null,
      "userPreferences": {
        "preferredTimeZone": "(UTC+00:00) UTC",
        "colorTheme": "NORMAL_CISCO_IDENTITY",
        "type": "userpreferences"
      },
      "userRole": "ROLE_READ_WRITE",
      "identitySourceId": "0a7996ae-6e5b-11e8-bd65-dbab801c44b9",
      "userServiceTypes": [
        "MGMT"
      ],
      "id": "29b20e67-6e64-11e8-bd65-3582e0f59b48",
      "type": "user",
      "links": {
        "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
object/users/29b20e67-6e64-11e8-bd65-3582e0f59b48"
      }
    }
  ],
  "paging": {
    "prev": [],
    "next": [],
    "limit": 10,
    "offset": 0,
    "count": 3,
    "pages": 0
  }
}
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。