Configurer des utilisateurs externes pour l'API

Exigence relative à la version : pour utiliser AAA externe, vous devez exécuter la version 6.3(0) ou supérieure de défense contre les menaces, et la version v2 ou supérieure de l'API REST défense contre les menaces.

Vous pouvez configurer le dispositif pour qu'il utilise un serveur RADIUS AAA externe pour authentifier et autoriser l'accès des utilisateurs à l'API REST défense contre les menaces. Vous pouvez utiliser les comptes d'utilisateur RADIUS à la place ou en plus du compte d'utilisateur **admin** local intégré.

Lorsque vous utilisez un serveur AAA externe, vous pouvez définir différents niveaux d'autorisation pour différents comptes. Cela vous permet de limiter les entités qui peuvent apporter des modifications à la configuration du dispositif tout en fournissant un accès en lecture seule au personnel de soutien.

La procédure suivante explique le processus de bout en bout de configuration des comptes RADIUS et de configuration du dispositif pour qu'il utilise un protocole AAA externe pour l'authentification et l'autorisation.

Avant de commencer

Gardez les facteurs opérationnels suivants à l'esprit lorsque vous utilisez une autorisation externe.

- Si le dispositif est configuré pour fonctionner en mode haute disponibilité, configurez l'autorisation externe sur l'unité active. Vous devez ensuite exécuter une tâche de déploiement pour les paramètres d'autorisation afin de permettre l'accès utilisateur au dispositif de secours.
- Chaque fois qu'un nouvel utilisateur accède au système, une ressource d'utilisateur est créée pour ce dernier. Vous devez déployer la configuration pour enregistrer cet objet utilisateur.

  (Versions antérieures à 6.6. de défense contre les menaces) Si vous travaillez en mode haute disponibilité (HA), vous devez déployer la configuration avant que cet utilisateur puisse se connecter à l'unité de secours. Étant donné que seuls les utilisateurs administrateurs ou en lecture-écriture peuvent démarrer une tâche de déploiement, un nouvel utilisateur en lecture seule doit demander à une autre personne de déployer la configuration pour enregistrer l'objet utilisateur.

  À partir de la version 6.6 de défense contre les menaces, les restrictions du mode HA sont supprimées. Un utilisateur externe peut se connecter à l'unité en veille sans d'abord se connecter à l'unité active et déployer la configuration. L'objet utilisateur ne sera pas créé sur l'unité en veille, mais les caractéristiques de l'utilisateur seront mises en mémoire cache et l'accès sera accordé à l'utilisateur, en supposant qu'un nom d'utilisateur et un mot de passe valides sont fournis.

Procédure

Étape 1 Définir les droits d'autorisation dans les comptes d'utilisateurs RADIUS, à la page 2.

Étape 2 Définir les serveurs RADIUS, à la page 3.

Étape 3 Créer un groupe de serveurs AAA pour les serveurs RADIUS, à la page 4.

Étape 4 Désigner le groupe de serveurs AAA comme source d'authentification pour l'accès HTTPS, à la page 7.

Étape 5 Utilisez **POST /operational/deploy** pour démarrer une tâche de déploiement.

La commande **curl** ressemblerait à ce qui suit :

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

'https://ftd.example.com/api/fdm/dernière version/operational/deploy'
```

Pour de plus amples renseignements, consultez Déploiement des modifications de configuration.

Étape 6 Vérifier l'accès d'utilisateurs externes, à la page 10.

Définir les droits d'autorisation dans les comptes d'utilisateurs RADIUS

Vous pouvez fournir un accès SSH à l'API REST de défense contre les menacesà partir d'un serveur RADIUS externe. En activant l'authentification et l'autorisation RADIUS, vous pouvez fournir différents niveaux de droits d'accès et ne pas demander à chaque utilisateur de se connecter avec le compte **admin** local.

Remarque Ces utilisateurs externes sont également autorisés pour gestionnaire d'appareil.

Pour fournir un contrôle d'accès basé sur les rôles (RBAC), mettez à jour les comptes d'utilisateur sur votre serveur RADIUS pour définir l'attribut **cisco-av-pair**. Cet attribut doit être défini correctement sur un compte utilisateur, sinon l'utilisateur se voit refuser l'accès à l'API REST. Voici les valeurs suivantes prises en charge pour l'attribut cisco-av-pair :

- **fdm.userrole.authority.admin** fournit un accès administrateur complet. Ces utilisateurs peuvent effectuer toutes les actions que l'utilisateur **admin** local peut effectuer.
- **fdm.userrole.authority.rw** fournit un accès en lecture-écriture. Ces utilisateurs peuvent faire tout ce qu'un utilisateur en lecture seule peut faire, ainsi que modifier et déployer la configuration. Les seules restrictions concernent les actions critiques pour le système, qui comprennent l'installation des mises à

niveau, la création et la restauration de sauvegardes, l'affichage du journal d'audit et la déconnexion d'autres utilisateurs.

- **fdm.userrole.authority.ro** fournit un accès en lecture seule. Ces utilisateurs peuvent afficher les tableaux de bord et la configuration, mais ne peuvent apporter aucune modification. Si l'utilisateur tente d'apporter une modification, le message d'erreur explique que cela est causé par un manque d'autorisation.

Définir les serveurs RADIUS

Après avoir configuré les comptes d'utilisateur dans le serveur RADIUS pour définir les droits d'autorisation appropriés, vous pouvez configurer le dispositif pour qu'il utilise le serveur pour authentifier et autoriser l'accès à l'API REST.

Utilisez la ressource **POST /object/radiusidentitysources** pour créer un objet pour chaque serveur RADIUS que vous souhaitez définir.

Procédure

Étape 1 Créez le corps de l'objet JSON pour le serveur RADIUS.

Voici un exemple d'objet JSON à utiliser avec cet appel.

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

Les attributs sont :

- **name** (nom) : le nom de l'objet. Cela ne doit pas absolument correspondre à tout ce qui est défini dans le serveur RADIUS.
- **description** (description) : (facultatif.) Une description de l'objet.
- **host** (domaine) : l'adresse IP ou le nom de domaine complet du serveur RADIUS.
- **timeout** (délai d'expiration) : (facultatif.) La durée, de 1 à 300 secondes, pendant laquelle le système attend une réponse du serveur avant d'envoyer la demande au serveur suivant. Si vous n'incluez pas cet attribut, la valeur par défaut est de 10 secondes.
- **serverAuthenticationPort** (portD'AuthentificationDuServeur) : (facultatif.) Le port sur lequel l'authentification et l'autorisation RADIUS sont effectuées. Si vous n'incluez pas cet attribut, la valeur par défaut sera 1812.
- **serverSecretKey** (cléSecrèteDuServeur) : (facultatif.) le code secret partagé qui est utilisé pour chiffrer les données entre le dispositif défense contre les menaceset le serveur RADIUS. La clé est une chaîne de caractères alphanumériques sensible à la casse et composé de jusqu'à 64 caractères, espaces exclus. La clé doit commencer par un caractère alphanumérique ou un trait de soulignement et peut contenir les

caractères spéciaux : $ & - _ . + @. Cette chaîne de caractères doit correspondre à la clé configurée sur le serveur RADIUS. Si vous ne configurez pas de clé secrète, la connexion ne sera pas chiffrée.

Étape 2 Postez l'objet.

Par exemple, la commande **curl** ressemblerait à ce qui suit :

```
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/dernière version/object/radiusidentitysources'
```

Étape 3 Vérifiez la réponse.

Vous devriez obtenir un code réponse de 200. Un corps de réponse qui fonctionne ressemblera à ce qui suit : Notez que les informations sensibles, telles que la clé secrète, sont masquées dans la réponse.

```
{
  "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/dernière version/object/
radiusidentitysources/1b962e3b-6e56-11e8-bd65-379fa8aaaba1"
  }
}
```

Créer un groupe de serveurs AAA pour les serveurs RADIUS

Après avoir créé les objets de serveur RADIUS, utilisez la ressource **POST /object/radiusidentitysourcegroups** pour créer un groupe AAA contenant les objets radiusidentitysource.

Vous pouvez ajouter jusqu'à 16 serveurs RADIUS à un groupe de serveurs RADIUS AAA. Ces serveurs doivent être des copies de sauvegarde les uns des autres, c'est-à-dire qu'ils doivent avoir la même liste de comptes d'utilisateurs.

Procédure

Étape 1 Créez le corps de l'objet JSON pour le groupe de serveurs RADIUS.

Voici un exemple d'objet JSON à utiliser avec cet appel.

```
{
  "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"
}
```

Les attributs sont :

- **name** (nom) : le nom de l'objet. Cela ne doit pas obligatoirement correspondre à tout ce qui est défini dans les serveurs RADIUS du membre.
- **maxFailedAttempts** (nombreMaximumDeTentativesÉchouées) : (facultatif.) Les serveurs défaillants ne sont réactivés que lorsque tous les serveurs sont tombés en panne. Le temps mort est le temps d'attente, de 0 à 1440 minutes, après l'échec du dernier serveur avant la réactivation de tous les serveurs. Si vous n'incluez pas cet attribut, la valeur par défaut sera 10 minutes.
- **deadTime** (tempsMort) : (facultatif.) Le nombre de demandes ayant échoué (c'est-à-dire de demandes qui ne reçoivent pas de réponse) envoyées à un serveur RADIUS du groupe avant d'essayer le serveur suivant. Vous pouvez spécifier de 1 à 5, et la valeur par défaut est 3. Lorsque le nombre maximal de tentatives ayant échoué est dépassé, le système marque le serveur comme ayant échoué.

  Pour une fonctionnalité donnée, si vous avez configuré une méthode de secours à l'aide de la base de données locale et qu'aucun serveur du groupe ne répond, le groupe est considéré comme ne répondant pas et la méthode de secours est essayée. Le groupe de serveurs reste marqué comme ne répondant pas pendant la durée du temps mort, de sorte que les demandes AAA supplémentaires effectuées dans cette période ne résultent pas en une tentative d'entrer en contact avec le groupe de serveurs et que la méthode de secours est utilisée immédiatement.
- **description** (description) : (facultatif.) Une description de l'objet.
- **radiusIdentitySources** (sourceD'IdentitéRadius) : il s'agit d'un groupe d'éléments qui définit chaque objet radiusidentitysource (sourced'identitéradius) qui définit un serveur RADIUS à inclure dans le groupe. Incluez les éléments entre les [crochets]. Voici les attributs et la syntaxe de chaque objet. Vous obtenez la valeur des attributs **id** (identifiant), **version** (version) et **name** (nom) des objets individuels; les informations se trouvent dans le corps de la réponse lorsque vous créez les objets. Vous pouvez également obtenir les informations à partir d'un appel **GET /object/radiusidentitysources**. Le **type** doit être **radiusidentitysource**.

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

Étape 2 Postez l'objet.

Par exemple, la commande **curl** ressemblerait à ce qui suit :

```
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/dernière version/object/radiusidentitysourcegroups'
```

Étape 3 Vérifiez la réponse.

Vous devriez obtenir un code réponse de 200. Un corps de réponse qui fonctionne ressemblera à ce qui suit :

```
{
  "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/dernière version/object/
radiusidentitysourcegroups/0a7996ae-6e5b-11e8-bd65-dbab801c44b9"
  }
}
```

Désigner le groupe de serveurs AAA comme source d'authentification pour l'accès HTTPS

Utilisez la ressource **PUT /devicesettings/default/aaasettings/{objId}** pour identifier le groupe de serveurs RADIUS AAA comme source d'identité pour l'autorisation de l'utilisateur.

Il n'y a pas de méthode POST : les objets nécessaires à l'authentification du système existent déjà. Vous devez d'abord faire appel à une méthode GET pour déterminer les valeurs d'identifiant et de version pertinentes.

Procédure

Étape 1 Utilisez **GET /devicesettings/default/aaasettings** pour déterminer les attributs des objets aaasettings.

La commande **curl** ressemblerait à ce qui suit :

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/dernière version/devicesettings/default/aaasettings'
```

Par exemple, le corps de la réponse devrait ressembler à ce qui suit. Cet exemple montre que la source d'identité locale est celle définie pour l'accès HTTPS. Il est également utilisé pour l'accès SSH, qui n'est pas pertinent pour l'API REST.


```
{
  "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/dernière version/
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/dernière version/
devicesettings/default/aaasettings/00000003-0000-0000-0000-000000000008"
      }
    }
  ],
  "paging": {
    "prev": [],
    "next": [],
    "limit": 10,
    "offset": 0,
    "count": 2,
    "pages": 0
  }
}
```

Étape 2 (Facultatif) Utilisez **GET /devicesettings/default/aaasettings/{objId}** pour obtenir une copie de l'objet de paramètre HTTPS AAA afin de restreindre votre vue.

Votre appel PUT mettra à jour l'objet HTTPS uniquement. Vous n'avez pas besoin de mettre à jour l'objet SSH.

Dans cet exemple, l'identifiant de l'objet HTTPS est 00000003-0000-0000-0000-000000000007. La commande **curl** ressemblerait donc à ce qui suit :

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/dernière version/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007'
```

Le corps de la réponse ressemblerait à ce qui suit.

```
{
  "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/dernière version/
devicesettings/default/aaasettings/00000003-0000-0000-0000-000000000007"
  }
```

Étape 3 Créez le corps d'objet JSON pour l'accès à la gestion AAA.

Voici un exemple d'objet JSON à utiliser avec cet appel.

```
{
  "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"
}
```

Les attributs sont :

- **version** (version) : la version de l'objet HTTPS. Vous trouverez cette valeur dans le corps de la réponse pour l'appel de la méthode GET.
- **name** (nom) : le nom de l'objet, **HTTPS**. Vous trouverez cette valeur dans le corps de la réponse pour l'appel de la méthode GET.
- **identitySourceGroup** (groupeSourceDeL'Identité) : cette valeur identifie le groupe de serveurs RADIUS. Obtenez les valeurs **id** (identifiant), **version** (version) et **name** (nom) dans le corps de la réponse une fois le groupe de serveurs (ou un appel **GET /object/radiusidentitysourcegroups**) créé. Le type doit être **radiusidentitysourcegroup**.
- **description** (description) : (facultatif.) Une description de l'objet.
- **protocolType** (typeDeProtocole) : le protocole auquel cette source s'applique, **HTTPS**.
- **useLocal** (utiliserLocal) : comment utiliser la source d'identité locale, qui contient le compte d'utilisateur administrateur local. Saisissez l'une des options suivantes :
  - **Before** (Avant) : le système vérifie d'abord le nom d'utilisateur et le mot de passe par rapport à la source locale.
  - **After** (Après) : la source locale est vérifiée uniquement si la source externe n'est pas disponible ou si le compte d'utilisateur n'a pas été trouvé dans la source externe.
  - **Never** (Jamais) : (non recommandé.) La source locale n'est jamais utilisée, vous ne pouvez donc pas vous connecter en tant qu'utilisateur **admin**.

    Mise en garde
    Si vous sélectionnez **Never** (Jamais), vous ne pourrez pas vous connecter au gestionnaire d'appareil ni utiliser l'API en utilisant le compte **admin**. Vous serez verrouillé du système si le serveur RADIUS n'est plus disponible ou si vous ne configurez pas les comptes correctement dans le serveur RADIUS.

- **id** : la valeur d'identifiant de l'objet HTTPS. Vous trouverez cette valeur dans le corps de la réponse pour l'appel de la méthode GET.
- **type** (type) : le type d'objet, **aaasetting**.

Étape 4 Placez l'objet.

Par exemple, la commande **curl** ressemblerait à ce qui suit. Notez que l'objet {objId} dans l'URL et l'identifiant de l'objet aaasettings dans l'objet JSON sont identiques.

```
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/dernière version/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007'
```

Étape 5 Vérifiez la réponse.

Vous devriez obtenir un code réponse de 200. Un corps de réponse qui fonctionne ressemblera à ce qui suit :

```
{
  "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/dernière version/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007"
  }
}
```

Vérifier l'accès d'utilisateurs externes

Une fois la tâche de déploiement terminée, vous pouvez tester l'accès d'un utilisateur externe au gestionnaire d'appareil et à l'API REST.

Procédure

Étape 1 Connectez-vous au gestionnaire d'appareil en utilisant un nom d'utilisateur externe doté d'un attribut cisco-av-pair valide.

La connexion devrait réussir, et le coin supérieur droit de la page devrait afficher le nom d'utilisateur et le niveau de privilège.

Étape 2 Obtenez un jeton API REST pour un utilisateur externe.

Si l'utilisateur peut obtenir un jeton, il peut utiliser les ressources et les méthodes autorisées pour le niveau de privilège qui lui a été attribué.

a) Créez le corps de l'objet JSON pour un jeton simple attribué par mot de passe.

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

b) Utilisez **POST /fdm/token** pour obtenir le jeton.

```
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/dernière version/fdm/token'
```

c) Évaluez la réponse pour vérifier qu'un jeton a été accordé.

Vous devriez obtenir un code réponse de 200. Lorsque vous obtenez un jeton, cela signifie que le système a pu authentifier l'utilisateur.

```
{
  "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
}
```

Étape 3 Utilisez **GET /objet/users** pour vérifier que les objets utilisateur ont été créés pour chaque utilisateur.

Les objets utilisateur sont automatiquement créés pour chaque nouvel utilisateur qui se connecte au gestionnaire d'appareil ou qui obtient un jeton d'accès. Vous devez exécuter une tâche de déploiement pour enregistrer ces objets utilisateur. En mode haute disponibilité, la tâche de déploiement est requise avant que l'utilisateur puisse se connecter à l'unité en veille.

Par exemple, la commande **curl** ressemblerait à ce qui suit :

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/dernière version/object/users'
```

Le corps de réponse suivant indique que deux utilisateurs externes se sont connectés. Notez que la valeur **userRule** affiche les droits concédés par la commande cisco-av-pair et configurés dans le serveur RADIUS

pour ces comptes d'utilisateurs. Utilisez ces informations pour vérifier que vous avez configuré les comptes d'utilisateur du serveur RADIUS correctement. L'utilisateur **admin** est l'utilisateur défini localement.


```
{
  "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/dernière version/
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/dernière version/
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/dernière version/
object/users/29b20e67-6e64-11e8-bd65-3582e0f59b48"
      }
    }
  ],
  "paging": {
    "prev": [],
    "next": [],
    "limit": 10,
    "offset": 0,
    "count": 3,
    "pages": 0
  }
}
```

À propos de la traduction

Cisco peut fournir des traductions du présent contenu dans la langue locale pour certains endroits. Veuillez noter que des traductions sont fournies à titre informatif seulement et, en cas d’incohérence, la version anglaise du présent contenu prévaudra.