# API에 대한 외부 사용자 구성

버전 요구 사항: 외부 AAA를 사용하려면 threat defense 버전 6.3(0) 이상 및 threat defense REST API v2 이상을 실행해야 합니다.

외부 RADIUS AAA 서버를 사용하여 threat defense REST API에 대한 사용자 액세스를 인증하고 권한을 부여하도록 디바이스를 구성할 수 있습니다. 내장된 로컬 관리 사용자 어카운트 대신 또는 해당 어카운트와 함께 RADIUS 사용자 어카운트를 사용할 수 있습니다.

외부 AAA 사용할 때 각기 다른 권한 부여 레벨을 사용하도록 어카운트를 정의할 수 있습니다. 그러면 디바이스 컨피그레이션을 변경할 수 있는 사용자를 제한하는 동시에 지원 담당자에게는 읽기 전용 액세스 권한을 계속 제공할 수 있습니다.

다음 절차에서는 RADIUS 어카운트를 설정한 다음 인증과 권한 부여에 외부 AAA를 사용하도록 디바이스를 구성하는 전체 프로세스를 설명합니다.

시작하기 전에

외부 권한 부여를 사용할 때는 다음의 작동 요소에 유의해야 합니다.

- 디바이스의 고가용성이 구성된 경우에는 액티브 유닛에서 외부 권한 부여를 구성합니다. 그런 다음 권한 부여 설정에 대한 구축 작업을 실행하여 스탠바이 디바이스에 대한 사용자 액세스를 허용해야 합니다.
- 새 사용자가 시스템에 액세스할 때마다 해당 사용자에 대한 사용자 리소스가 생성됩니다. 해당 사용자 개체를 저장하려면 컨피그레이션을 구축해야 합니다.

  (threat defense 6.6 이전 버전) HA(고가용성) 모드로 작동하는 경우에는 구성을 구축해야 해당 사용자가 스탠바이 유닛에 로그인할 수 있습니다. 관리자 또는 읽기-쓰기 사용자만 구축 작업을 시작할 수 있으므로 최초 읽기 전용 사용자는 다른 사용자가 컨피그레이션을 구축하여 사용자 개체를 저장하도록 해야 합니다.

  threat defense 6.6부터는 HA 제한 사항이 제거됩니다. 외부 사용자는 먼저 액티브 유닛에 로그인하고 구성을 구축하지 않아도 스탠바이 유닛에 로그인할 수 있습니다. 사용자 개체는 스탠바이 유닛에서 생성되지 않지만, 유효한 사용자 이름/비밀번호가 제공되었다고 가정할 경우 사용자에게는 액세스 권한이 부여되며 사용자 특성은 캐시됩니다.

프로시저

단계 1 RADIUS 사용자 어카운트에 대한 권한 부여 권한 정의, 2 페이지.

단계 2 RADIUS 서버 정의, 3 페이지.

단계 3 RADIUS 서버용 AAA 서버 그룹 생성, 4 페이지.

단계 4 HTTP 액세스를 위한 인증 소스로 AAA 서버 그룹 설정, 6 페이지.

단계 5 **POST /operational/deploy**를 사용하여 구축 작업을 시작합니다.

**curl** 명령은 다음과 같습니다.

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

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

변경 사항 구축에 대한 자세한 내용은 컨피그레이션 변경 사항 구축의 내용을 참조하십시오.

단계 6 외부 사용자 액세스 확인, 10 페이지.



# RADIUS 사용자 어카운트에 대한 권한 부여 권한 정의

외부 RADIUS 서버에서 threat defense REST API에 액세스할 수 있는 권한을 제공할 수 있습니다. RADIUS 인증 및 권한 부여를 활성화하면 각기 다른 액세스 권한 레벨을 제공할 수 있으며, 모든 사용자가 로컬 관리자 어카운트를 통해 로그인하지 못하게 할 수 있습니다.

참고 이러한 외부 사용자에게 device manager에 대한 권한도 부여할 수 있습니다.

RBAC(Role-Based Access Control)를 제공하려면 RADIUS 서버에서 사용자 어카운트를 업데이트하여 **cisco-av-pair** 특성을 정의합니다. 사용자 어카운트에 대해 이러한 특성을 정확하게 정의해야 합니다. 그렇지 않으면 해당 사용자의 REST API 액세스가 거부됩니다. cisco-av-pair 특성에 대해 지원되는 값은 다음과 같습니다.

- **fdm.userrole.authority.admin**은 전체 관리자 액세스를 제공합니다. 이러한 사용자는 로컬 관리 사용자가 수행할 수 있는 모든 작업을 수행할 수 있습니다.
- **fdm.userrole.authority.rw**는 읽기-쓰기 액세스를 제공합니다. 이러한 사용자는 읽기 전용 사용자가 수행할 수 있는 모든 작업을 수행할 수 있으며 컨피그레이션 수정 및 구축도 수행할 수 있

습니다. 업그레이드 설치, 백업 생성 및 복원, 감사 로그 확인, 다른 사용자 로그오프를 포함하는 시스템의 중요 작업만 제한됩니다.

- **fdm.userrole.authority.ro**는 읽기 전용 액세스를 제공합니다. 이러한 사용자는 대시보드 및 컨피그레이션을 볼 수는 있지만 변경할 수는 없습니다. 사용자가 변경을 시도하면 권한이 없음을 설명하는 오류 메시지가 표시됩니다.

# RADIUS 서버 정의

적절한 권한 부여 권한을 정의하도록 RADIUS 서버의 사용자 어카운트를 구성한 후에는 서버를 사용하여 REST API에 대한 액세스를 인증하고 권한을 부여하도록 디바이스를 구성할 수 있습니다.

**POST /object/radiusidentitysources** 리소스를 사용하여 정의하려는 각 RADIUS 서버용 개체를 생성합니다.

프로시저

단계 **1** RADIUS 서버용 JSON 개체 본문을 생성합니다.

이 호출에 사용할 JSON 개체의 예는 다음과 같습니다.

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

특성은 다음과 같습니다.

- **name(**이름**)** - 개체 이름입니다. 이 이름은 RADIUS 서버에 정의된 항목과 일치하지 않아도 됩니다.
- **description(**설명**)** - (선택 사항). 개체의 설명입니다.
- **host(**호스트**)** - RADIUS 서버의 IP 주소 또는 정규화된 호스트 이름입니다.
- **timeout(**시간 제한**)** - (선택 사항). 시스템이 다음 서버로 요청을 보내기 전까지 서버의 응답을 기다리는 시간(1~300초)입니다. 이 특성을 포함하지 않는 경우의 기본값은 10초입니다.
- **serverAuthenticationPort** - (선택 사항). RADIUS 인증 및 권한 부여가 수행되는 포트입니다. 이 특성을 포함하지 않는 경우의 기본값은 1812입니다.
- **serverSecretKey** - (선택 사항). threat defense 디바이스와 RADIUS 서버 간의 데이터를 암호화하는 데 사용되는 공유 암호입니다. 이 키는 대/소문자를 구분하며 공백은 포함하지 않는 영숫자 문자열(최대 64자)입니다. 또한 영숫자 문자 또는 밑줄로 시작해야 하며 특수 문자 $ & - _ . + @

는 포함할 수 없습니다. 문자열은 RADIUS 서버에 구성된 것과 일치해야 합니다. 비밀 키를 구성하지 않으면 연결이 암호화되지 않습니다.

단계 **2** 개체를 게시합니다.

예를 들어 **curl** 명령은 다음과 같이 표시됩니다.

```
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/최신/object/radiusidentitysources'
```

단계 **3** 응답을 확인합니다.

응답 코드 200이 표시되어야 합니다. 올바른 응답 본문은 다음과 같이 표시됩니다. 비밀 키와 같은 민감한 정보는 응답에서 마스크 처리됩니다.

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

# RADIUS 서버용 AAA 서버 그룹 생성

RADIUS 서버 개체를 생성한 후에는 **POST /object/radiusidentitysourcegroups** 리소스를 사용하여 radiusidentitysource 개체를 포함할 AAA 그룹을 생성합니다.

RADIUS AAA 서버 그룹에는 RADIUS 서버를 16개까지 추가할 수 있습니다. 이러한 서버는 서로의 백업이어야 합니다. 즉, 서버의 사용자 어카운트 목록이 같아야 합니다.

프로시저

단계 **1** RADIUS 서버 그룹용 JSON 개체 본문을 생성합니다.

이 호출에 사용할 JSON 개체의 예는 다음과 같습니다.

```
{
  "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"
}
```

특성은 다음과 같습니다.

- **name(**이름**)** - 개체 이름입니다. 이 이름은 멤버 RADIUS 서버에 정의된 항목과 일치하지 않아도 됩니다.
- **maxFailedAttempts** - (선택 사항). 실패한 서버는 모든 서버가 실패한 후에야 비로소 재활성화됩니다. 비활성 시간이란 마지막 서버가 실패한 이후, 모든 서버를 재활성화하기 이전의 대기 시간(0~1440분)입니다. 이 특성을 포함하지 않는 경우 기본값은 10분입니다.
- **deadTime** - (선택 사항). 다음 서버 사용을 시도하기 전에 그룹의 RADIUS 서버로 전송되었으나 실패한 요청(즉, 응답을 받지 못한 요청)의 수입니다. 1~5 사이의 값을 지정할 수 있으며 기본값은 3입니다. 최대 실패 시도 횟수가 초과되면 시스템에서 해당 서버를 Failed(장애 발생)로 표시합니다.

  특정 기능에 대해 로컬 데이터베이스를 사용하여 대체 방법을 구성했는데 그룹의 모든 서버가 응답하지 않으면 해당 그룹은 응답이 없는 것으로 간주되고 대체 방법을 시도합니다. 서버 그룹은 비활성 시간 동안 응답이 없는 것으로 표시됩니다. 따라서 이 기간 내에는 추가 AAA 요청이 서버 그룹 연결을 시도하지 않으며 대체 방법이 즉시 사용됩니다.
- **description(**설명**)** - (선택 사항). 개체의 설명입니다.
- **radiusIdentitySources** - 그룹에 포함할 RADIUS 서버를 정의하는 각 radiusidentitysource 개체를 정의하는 항목의 그룹입니다. [괄호] 안에 항목을 포함합니다. 아래에는 각 개체의 특성과 구문이 나와 있습니다. 개별 개체에서 **id**, **version(**버전**)** 및 **name(**이름**)** 속성의 값을 가져옵니다. 개체 생성 시 정보는 응답 본문에 있습니다. **GET /object/radiusidentitysources** 호출에서 정보를 가져올 수도 있습니다. **type(**유형**)**은 **radiusidentitysource**이어야 합니다.

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

```
        "name": "aaa-server-1"
      }
```

단계 **2** 개체를 게시합니다.

예를 들어 **curl** 명령은 다음과 같이 표시됩니다.

```
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/최신/object/radiusidentitysourcegroups'
```

단계 **3** 응답을 확인합니다.

응답 코드 200이 표시되어야 합니다. 올바른 응답 본문은 다음과 같이 표시됩니다.

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

# HTTP 액세스를 위한 인증 소스로 AAA 서버 그룹 설정

**PUT /devicesettings/default/aaasettings/{objId}** 리소스를 사용하여 RADIUS AAA 서버 그룹을 사용자 인증용 ID 소스로 식별합니다.

POST 메서드는 없으며 시스템 인증에 필요한 개체는 이미 있습니다. 먼저 GET을 수행하여 관련 ID 및 버전 값을 확인해야 합니다.

프로시저

단계 **1** **GET /devicesettings/default/aaasettings**를 사용하여 aaasettings 개체의 특성을 확인합니다.

**curl** 명령은 다음과 같습니다.

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

예를 들어 응답 본문은 다음과 같습니다. 이 예시에서는 로컬 ID 소스가 HTTPS 액세스용으로 정의된 소스임을 보여줍니다. 이 소스는 SSH 액세스(REST API와는 관련이 없음)에도 사용됩니다.

```
{
  "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/최신/
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/최신/
devicesettings/default/aaasettings/00000003-0000-0000-0000-000000000008"
      }
    }
  ],
  "paging": {
    "prev": [],
```

```
    "next": [],
    "limit": 10,
    "offset": 0,
    "count": 2,
    "pages": 0
  }
}
```

단계 **2** (선택 사항). **GET /devicesettings/default/aaasettings/{objId}**를 사용하여 HTTPS AAA 설정 개체 복사본을 가져와 보기의 범위를 좁힙니다.

PUT 호출에서는 HTTPS 개체만 업데이트됩니다. SSH 개체는 업데이트할 필요가 없습니다.

이 예시에서 HTTPS 개체의 ID는 00000003-0000-0000-0000-000000000007이므로 **curl** 명령은 다음과 유사합니다.

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

응답 본문은 다음과 유사합니다.

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

단계 **3** AAA 관리 액세스용 JSON 개체 본문을 생성합니다.

이 호출에 사용할 JSON 개체의 예는 다음과 같습니다.

```
{
  "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"
}
```

특성은 다음과 같습니다.

- **version(**버전**)** - HTTPS 개체의 버전입니다. GET 호출에 대한 응답 본문에서 이 값을 찾을 수 있습니다.
- **name(**이름**)** - 개체 이름(**HTTPS**)입니다. GET 호출에 대한 응답 본문에서 이 값을 찾을 수 있습니다.
- **identitySourceGroup** - RADIUS 서버 그룹을 식별하는 특성입니다. 서버 그룹 또는 **GET /object/radiusidentitysourcegroups** 호출을 생성한 경우, 응답 본문에서 **id**, **version(**버전**)** 및 **name(**이름**)** 값을 가져옵니다. 유형은 **radiusidentitysourcegroup**이어야 합니다.
- **description(**설명**)** - (선택 사항). 개체의 설명입니다.
- **protocolType** - 이 소스가 적용되는 프로토콜(**HTTPS**)입니다.
- **useLocal** - 로컬 관리 사용자 어카운트가 포함된 로컬 ID 소스를 사용할 방법입니다. 다음 옵션 중 하나를 입력합니다.
  - **Before(**전**)** - 시스템이 로컬 소스를 먼저 대조하여 사용자 이름과 비밀번호를 확인합니다.
  - **After(**후**)** - 외부 소스를 사용할 수 없거나 외부 소스에 사용자 어카운트가 없는 경우에만 로컬 소스를 확인합니다.
  - **Never(**사용 안 함**)** — (권장되지 않음.) 로컬 소스를 사용하지 않습니다. 따라서 관리 사용자로 로그인할 수 없습니다.

    주의 **Never(**사용 안 함**)**를 선택하는 경우 관리자 어카운트를 사용하여 device manager에 로그인하거나 API를 사용할 수 없습니다. RADIUS 서버를 사용할 수 없게 되거나 RADIUS 서버에서 어카운트를 잘못 구성하는 경우에는 시스템에서 해당 어카운트를 차단합니다.

- **id** - HTTPS 개체의 ID 값입니다. GET 호출에 대한 응답 본문에서 이 값을 찾을 수 있습니다.
- **type(**유형**)** - 개체 유형(**aaasetting**)입니다.

단계 **4** 개체를 배치합니다.

예를 들어 **curl** 명령은 다음과 같습니다. URL의 {objId}와 JSON 개체의 aaasettings 개체 id는 같습니다.

```
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/최신/devicesettings/
default/aaasettings/00000003-0000-0000-0000-000000000007'
```

단계 **5** 응답을 확인합니다.

응답 코드 200이 표시되어야 합니다. 올바른 응답 본문은 다음과 같이 표시됩니다.

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

# 외부 사용자 액세스 확인

구축 작업이 완료되고 나면 device manager와 REST API 모두에 대한 외부 사용자 액세스를 테스트할 수 있습니다.

프로시저

단계 **1** 유효한 cisco-av-pair 특성이 포함된 외부 사용자 이름을 사용하여 device manager에 로그인합니다.

로그인이 성공해야 하며 페이지 오른쪽 상단에 사용자 이름과 권한 레벨이 표시되어야 합니다.

단계 **2** 외부 사용자에 대한 REST API 토큰을 획득합니다.

토큰을 획득할 수 있는 사용자는 할당된 권한 레벨에 대해 허용되는 리소스와 메서드를 사용할 수 있습니다.

a) 단순 비밀번호 부여 토큰용 JSON 개체 본문을 생성합니다.

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

b) **POST /fdm/token**을 사용하여 토큰을 획득합니다.

```
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/최신/fdm/token'
```

c) 응답을 평가하여 토큰이 부여되었는지 확인합니다.

응답 코드 200이 표시되어야 합니다. 토큰을 획득했다는 것은 시스템에서 사용자를 인증할 수 있었다는 의미입니다.

```
{
  "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
}
```

단계 **3** **GET /object/users**를 사용하여 각 사용자에 대해 사용자 개체가 생성되었는지 확인합니다.

사용자 개체는 device manager에 로그인하거나 액세스 토큰을 획득하는 각 새 사용자에 대해 자동으로 생성됩니다. 이러한 사용자 개체를 저장하려면 구축 작업을 실행해야 합니다. 고가용성 모드에서는 구축 작업을 실행해야 사용자가 스탠바이 유닛에 로그인할 수 있습니다.

예를 들어 **curl** 명령은 다음과 같이 표시됩니다.

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

다음 응답 본문에는 두 외부 사용자가 로그인했음이 표시됩니다. **userRole**은 이러한 사용자 어카운트에 대해 RADIUS 서버에 구성된 cisco-av-pair에서 획득한 권한을 표시합니다. 이 정보를 사용하여 RADIUS 사용자 어카운트를 정확하게 구성했는지 확인합니다. 관리 사용자는 로컬에 정의된 사용자입니다.

```
{
  "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/최신/
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/최신/
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/최신/
object/users/29b20e67-6e64-11e8-bd65-3582e0f59b48"
      }
    }
```

```
  ],
  "paging": {
    "prev": [],
    "next": [],
    "limit": 10,
    "offset": 0,
    "count": 3,
    "pages": 0
  }
}
```