구성 가져오기/내보내기

버전 요구 사항: 구성 가져오기/내보내기를 사용하려면 위협 방어 6.5(0) 이상 버전 및 위협 방어 REST API v4 이상 버전을 실행해야 합니다.

device manager로 관리되는 디바이스에서 구성을 내보내고 구성을 동일한 디바이스 또는 다른 호환 가능한 디바이스로 가져올 수 있습니다. 예를 들어 구성 가져오기/내보내기를 사용하여 여러 유사한 디바이스 전반에서 베이스라인 구성을 복제한 다음, 각 디바이스의 device manager을 사용하여 각 디바이스에 고유한 특성을 구성할 수 있습니다.

컨피그레이션 가져오기/내보내기 정보

device manager을 사용하거나 CDO를 통해 위협 방어 디바이스를 로컬로 관리하는 경우, 위협 방어 API를 사용하여 디바이스의 구성을 내보낼 수 있습니다. 이 메서드는 Secure Firewall Management Center에서 관리하는 디바이스에서 효과가 없습니다.

구성을 내보내면 zip 파일이 생성됩니다. 그러면 zip 파일을 워크스테이션에 다운로드할 수 있습니다. 이 구성 자체는 JSON 형식 텍스트 파일에서 특성-값 쌍을 사용하여 정의된 개체로 표시됩니다. 동일한 디바이스 또는 다른 디바이스로 파일을 다시 가져오기 전에 파일을 수정할 수 있습니다.

따라서 내보내기 파일을 사용하여 네트워크의 다른 디바이스에 구축 가능한 템플릿을 생성할 수 있습니다.

개체를 가져올 때, 구성 파일이 아닌 import 명령으로 개체를 직접 정의하는 옵션을 사용할 수도 있습니다. 그러나 적은 수의 변경 사항을 가져오는 경우에만 개체를 직접 정의해야 합니다.

다음 주제에서는 구성 가져오기/내보내기에 대해 자세히 설명합니다.

내보내기 파일에 포함된 내용

내보내기를 수행할 때 내보내기 파일에 포함할 구성을 지정합니다. 전체 내보내기의 경우, 내보내기 zip 파일에 모든 항목이 포함됩니다. 내보내기 위해 선택한 항목에 따라 내보내기 zip 파일에는 다음 항목이 포함될 수 있습니다.

- 구성된 각 개체를 정의하는 특성-값 쌍. 구성 가능한 모든 항목은 device manager에서 "개체"라고 부르는 항목이 아니라 개체로서 모델링됩니다.
- 원격 액세스 VPN을 구성한 경우, AnyConnect 패키지 및 기타 참조된 모든 파일(예: 클라이언트 프로필 XML 파일, DAP XML 파일, Hostscan 패키지)
- 맞춤형 파일 정책을 구성한 경우, 모든 참조된 정리 목록 또는 맞춤형 탐지 목록

가져오기/내보내기 및 백업/복원 비교

구성 가져오기/내보내기는 백업/복원과 동일하지 않습니다.

- 백업/복원은 재해 복구를 위한 작업입니다. 디바이스가 동일한 모델이며 백업이 수행된 디바이스와 동일한 소프트웨어 버전을 실행 중인 경우에만 디바이스에 백업을 복원할 수 있습니다. 이는 주로 동일한 디바이스에 대해 "마지막으로 양호했던" 구성을 복구하거나, 구성을 대체 디바이스에 복원하기 위한 것입니다.
- 가져오기/내보내기는 구성의 전체 또는 일부를 유지하기 위한 것입니다. 디바이스를 이미지로 다시 설치한 후에 내보내기 파일을 사용하여 구성을 디바이스에 복원할 수 있습니다. 또는 내보내기 파일을 템플릿으로 사용하여 다른 디바이스로 가져오기 전에 콘텐츠를 수정할 수 있습니다. 가져오기/내보내기를 사용하면 새 디바이스를 특정 베이스라인 구성으로 빠르게 가져올 수 있으므로 이를 네트워크에 더 빠르게 구축할 수 있습니다. 제한 이내에서 파일을 다른 디바이스 모델(예: Firepower 2120~2130)로 가져올 수도 있습니다. 가져오기 파일에 모든 디바이스 모델에서 지원되는 개체만 포함되는 경우, 가져오기에 대한 제한 사항이 거의 없어야 합니다. 한 가지 제한 사항은 디바이스에서 내보내기 파일에 사용된 버전과 동일한 API 버전을 사용해야 한다는 것입니다.

가져오기/내보내기 전략

다음은 가져오기/내보내기를 사용할 수 있는 몇 가지 방법입니다.

- 새 디바이스의 템플릿을 생성합니다**.** 필요한 베이스라인에 맞게 모델 디바이스를 구성한 다음, 전체 구성을 내보냅니다. 그 후에는 해당 구성을 새 디바이스로 가져온 다음, device manager 또는위협 방어 API를 사용하여 필요한 수정 작업을 수행할 수 있습니다. 또한 가져오기 전에 템플릿을 수정하여 수정 작업을 수행할 수 있는데 예를 들어, 각 인터페이스의 IP 주소를 수정할 수 있습니다. 전체 내보내기에는 ManagementIP 개체(type=managementip)가 포함됩니다. 이때 대상 디바이스에서 관리 주소 및 게이트웨이를 이미 구성한 경우 새 디바이스에 대한 템플릿을 생성할 때 내보내기 파일에서 이 개체를 제거해야 하며, 그렇지 않으면 관리 주소 지정 정보를 덮어쓰게 됩니다.
- 한 디바이스의 구성 변경 사항을 유사한 디바이스에 구축합니다**.** 예를 들어, 디바이스 A의 구성을 수정할 때 몇 가지 새로운 네트워크 개체 및 액세스 제어 규칙을 생성합니다. 그런 다음 보류 중인 변경 사항을 내보내고 이러한 변경 사항을 디바이스 B로 가져올 수 있습니다. 두 디바이스에서 구성을 구축하면 디바이스에서 동일한 새 규칙을 실행하게 됩니다.

- 시스템을 이미지로 다시 설치한 후에 구성을 다시 적용합니다**.** 디바이스를 이미지로 다시 설치하면 구성이 지워집니다. 전체 구성을 처음 내보내는 경우에는 이미지로 다시 설치하기를 완료한 후에 이를 가져올 수 있습니다.
- 대상 구성을 적용합니다**.** 내보내기 파일을 수정하거나 수동으로 생성할 수도 있으므로 다른 디바이스로 가져오려는 개체를 제외하고 모든 개체를 제거할 수 있습니다. 예를 들어 네트워크 개체 집합을 포함하는 구성 파일을 생성하고 이 파일을 사용하여 동일한 네트워크 개체 그룹을 모든 위협 방어 디바이스에 가져올 수 있습니다.

구성 가져오기/내보내기에 대한 지침

- 내보내기 작업 중에는 시스템에서 구성 데이터베이스에 대한 쓰기 잠금을 유지합니다. 작업을 완료할 때까지 API 또는 device manager을 사용하여 구성을 변경할 수 없습니다. 그러나 내보내기 작업 중에 device manager의 구성을 보거나 API에서 GET 호출을 사용할 수는 있습니다.
- 가져오기 작업 중에는 시스템에서 구성 데이터베이스에 대한 읽기 및 쓰기 잠금을 둘 다 유지합니다. 작업을 완료할 때까지 API 또는 device manager을 사용하여 구성을 보거나 변경할 수 없습니다.
- 가져온 구성이 기존 구성에 추가됩니다. 디바이스의 구성을 지우고 가져온 구성으로 교체할 수 없습니다. 가져오기 전에 디바이스 구성을 재설정해야 하는 경우, 디바이스 CLI로 이동하여 **configure manager delete** 명령을 실행한 다음, **configure manager local** 명령을 실행할 수 있습니다. 관리 인터페이스 구성만 유지됩니다.
- 디바이스에서 파일에 포함된 메타데이터 개체의 apiVersion 특성에 정의된 것과 동일한 API 버전을 실행 중인 경우에만 파일을 디바이스로 가져올 수 있습니다.

컨피그레이션 가져오기 및 내보내기

가져오기/내보내기 프로세스는 로컬의 매니지드 디바이스에서 구성을 내보내는 작업부터 시작됩니다. 그런 다음 내보내기 파일을 다운로드하고 필요에 따라 수정하여 동일한 디바이스 또는 호환 가능한 디바이스에 업로드하기 전에 해당 파일을 수정할 수 있습니다. 다음 주제에서는 각 단계에 대해 설명합니다.

구성 내보내기

POST /action/configexport 메서드를 사용하여 구성 내보내기 작업을 생성하고 시작합니다.

프로시저

단계 **1** 내보내기 작업용 JSON 개체 본문을 생성합니다.

이 호출에 사용할 JSON 개체의 예는 다음과 같습니다.

```
{
  "diskFileName": "string",
  "encryptionKey": "*********",
  "doNotEncrypt": false,
  "configExportType": "FULL_EXPORT",
  "deployedObjectsOnly": true,
  "entityIds": [
    "string"
  ],
  "jobName": "string",
  "type": "scheduleconfigexport"
}
```

특성은 다음과 같습니다.

- **diskFileName** - (선택 사항) 내보내기 zip 파일의 이름입니다. 이름을 지정하지 않으면 시스템에서 대신 이름을 생성합니다. 이름을 지정하는 경우에도 시스템에서 고유성을 보장하기 위해 이름에 문자를 추가할 수 있습니다. 이름의 최대 길이는 60자입니다.
- **encryptionKey** - (선택 사항) zip 파일의 암호화 키입니다. 파일을 암호화하지 않으려면 이 필드를 생략하고 대신 "doNotEncrypt": true를 지정합니다. 키를 지정할 경우 zip 파일을 워크스테이션에 다운로드한 후에 해당 키를 사용하여 열어야 합니다. 내보낸 구성 파일에서는 비밀 키, 비밀번호 및 기타 중요한 데이터를 일반 텍스트에 표시합니다(다른 방법으로 가져올 수 없기 때문). 따라서 중요한 데이터를 보호하기 위해 암호화 키를 적용할 수 있습니다. 시스템에서 AES 256 암호화를 사용합니다.
- **doNotEncrypt** - (선택 사항) 내보내기 파일을 암호화할지(false), 암호화하지 않을지(true) 여부를 지정합니다. 기본값은 false입니다. 즉, 비어 있지 않은 encryptionKey 특성을 지정해야 합니다. true를 지정하면 encryptionKey 특성이 무시됩니다.
- **configExportType** - 다음 열거 값 중 하나입니다.
  - **FULL_EXPORT** - 전체 구성을 내보내기 파일에 포함합니다. 이는 기본값입니다.
  - **PARTIAL_EXPORT** - entityId 목록에서 식별된 개체 및 하위 개체만 포함합니다. 내보낼 수 없는 개체는 ID를 지정하는 경우에도 포함되지 않습니다. 모든 사용자 정의 개체는 내보낼 수 있습니다.
  - **PENDING_CHANGE_EXPORT** - 아직 구축되지 않은 개체 즉, 보류 중인 변경 사항만 포함합니다.
- **deployedObjectsOnly** - (선택 사항) 구축된 경우에만 내보내기 파일에 개체를 포함할지 여부입니다. 즉, 보류 중인 변경 사항은 포함하지 마십시오. 이 특성은 PENDING_CHANGE_EXPORT 작업의 경우 무시되는데 이러한 작업에는 구축되지 않은 개체만 포함되어 있기 때문입니다. 기본값은 false입니다. 이는 모든 보류 중인 변경 사항이 내보내기에 포함되었음을 의미합니다. 보류 중인 변경 사항을 제외하려면 true를 지정합니다.
- **entityIds** - 시작점 개체 집합의 ID가 쉼표로 구분된 목록으로,[대괄호]로 묶여 있습니다. 이 목록은 PARTIAL_EXPORT 작업에 필요합니다. 이 목록의 각 항목은 UUID 값 또는 "**id=***uuid-value*", "**type=***object-type*" 또는 "**name=***object-name*"과 같은 특성-값 쌍이 일치하는 패턴일 수 있습니다. 예를 들어 **"type=networkobject"**입니다.

type은 리프 엔터티(예: networkobject)이거나 리프 유형 집합의 별칭일 수 있습니다. 몇 가지 일반적인 유형의 별칭으로는 네트워크(NetworkObject 및 NetworkObjectGroup), 포트(모든 TCP/UDP/ICMP 포트, 프로토콜 및 그룹 유형), url(URL 개체 및 그룹), ikepolicy(IKE V1/V2 정책), ikeproposal(Ike V1/V2 제안), identitysource(모든 id 소스), 인증서(모든 인증서 유형), 개체(device manager에서 개체 페이지에 나열되는 모든 개체/그룹 유형), 인터페이스(모든 네트워크 인터페이스, s2svpn(모든 Site-to-Site VPN 관련 유형), ravpn(모든 RA VPN 관련 유형), vpn(s2svpn 및 ravpn 둘 다)이 있습니다.

이러한 모든 개체와 해당하는 발신 참조 하위 항목은 PARTIAL_EXPORT 출력 파일에 포함됩니다. 내보낼 수 없는 모든 개체는 ID를 지정한 경우에도 출력에서 제외됩니다. 적절한 리소스 유형에 대해 GET 메서드를 사용하여 대상 개체의 UUID, 유형 또는 이름을 가져옵니다.

예를 들어, 모든 네트워크 개체와 함께 myaccessrule이라는 액세스 규칙 및 UUID로 식별되는 개체 2개를 내보내려면 다음과 같이 지정하면 됩니다.

```
"entityIds": [
   "type=networkobject",
   "id=bab3e3cd-8c70-11e9-930a-1f12ee87d473",
   "name=myaccessrule",
   "acc2e3cd-8c70-11e9-930a-1f12ee87b286"
   ]",
```

- **jobName** - (선택 사항) 내보내기 작업의 이름입니다. 작업의 이름을 지정하면 작업 상태를 검색할 때 더 쉽게 찾을 수 있습니다.
- **type(**유형**)** - 작업 유형이며 항상 **scheduleconfigexport**입니다.

예제**:**

다음 예에서는 파일 export-config-1에 대한 전체 내보내기를 수행하고 기타 모든 특성에 대한 기본값을 수락합니다.

```
{
  "diskFileName": "export-config-1",
  "doNotEncrypt": true
  "configExportType": "FULL_EXPORT",
  "type": "scheduleconfigexport"
}
```

단계 **2** 개체를 게시합니다.

예를 들어 curl 명령은 다음과 같이 표시됩니다.

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
 -d '{ \
   "configExportType": "FULL_EXPORT", \
   "type": "scheduleconfigexport" \
 }' 'https://10.89.5.38/api/fdm/최신/action/configexport'
```

단계 **3** 응답을 확인합니다.

응답 코드 200이 표시되어야 합니다. 최소 JSON 개체를 게시한 경우, 성공적인 응답 본문은 다음과 같이 표시됩니다. 암호화 키를 지정하는 경우 응답에서 해당 키가 마스킹됩니다.

```
{
  "version": null,
  "scheduleType": "IMMEDIATE",
  "user": "admin",
  "forceOperation": false,
  "jobHistoryUuid": "c7a8ba61-629a-11e9-8b8d-0fcc3c9d6d0b",
  "ipAddress": "10.24.5.177",
  "diskFileName": "export-config-1",
  "encryptionKey": null,
  "doNotEncrypt": true
  "configExportType": "FULL_EXPORT",
  "deployedObjectsOnly": false,
  "entityIds": null,
  "jobName": "Config Export",
  "id": "c79be920-629a-11e9-8b8d-85231be77de0",
  "type": "scheduleconfigexport",
  "links": {
    "self": "https://10.89.5.38/api/fdm/최신
/action/configexport/c79be920-629a-11e9-8b8d-85231be77de0"
  }
}
```


내보내기 작업의 상태 확인

내보내기 작업을 완료하는 데는 시간이 약간 걸립니다. 구성할 항목이 많을수록 작업을 수행하는 데 더 많은 시간이 필요합니다. 작업 상태를 확인하여 파일 다운로드를 시도하기 전에 작업이 완료되었는지 확인합니다.

상태를 가져오는 가장 간단한 방법은 GET /jobs/configexportstatus를 사용하는 것입니다. 예를 들어 curl 명령은 다음과 같이 표시됩니다.

```
curl -X GET --header 'Accept: application/json'
'https://10.89.5.38/api/fdm/최신/jobs/configexportstatus'
```

완료된 작업의 경우 다음과 유사한 상태가 반환됩니다.


```
{
  "version": "hdy62yf5xp3vf",
  "jobName": "Config Export",
  "jobDescription": null,
  "user": "admin",
  "startDateTime": "2019-04-19 13:14:54Z",
  "endDateTime": "2019-04-19 13:14:56Z",
  "status": "SUCCESS",
  "statusMessage": "The configuration was exported successfully",
  "scheduleUuid": "1ef502ad-62a5-11e9-8b8d-074ebc750708",
  "diskFileName": "export-config-1.zip",
  "messages": [],
  "configExportType": "FULL_EXPORT",
  "deployedObjectsOnly": false,
  "entityIds": null,
  "id": "1f0aad8e-62a5-11e9-8b8d-bb1ebb4d1300",
  "type": "configexportjobstatus",
  "links": {
    "self": "https://10.89.5.38/api/fdm/최신
```

```
/jobs/configexportstatus/1f0aad8e-62a5-11e9-8b8d-bb1ebb4d1300"
  }
}
```

또는 GET /jobs/configexportstatus/{objId} 메서드를 사용하여 특정 작업에 대한 상태를 검색할 수 있습니다. 응답 개체의 **id** 필드에서 개체 ID를 가져옵니다.

내보내기 파일 다운로드

내보내기 작업이 완료되면 내보내기 파일이 시스템 디스크에 기록되며, 이 파일을 구성 파일이라고 합니다. GET /action/downloadconfigfile/{objId} 메서드를 사용하여 이 내보내기 파일을 워크스테이션에 다운로드할 수 있습니다. 사용 가능한 파일 목록을 가져오려면 GET /action/configfiles 메서드를 사용합니다.

참고 GET /action/downloadconfigfile/{objId}을 사용하여 일반적으로 파일 이름을 개체 ID로 지정합니다. 또는 파일과 관련된 ConfigExportStatus 개체의 ID를 지정할 수 있습니다.

프로시저

단계 **1** 디스크에 있는 구성 파일의 목록을 가져옵니다.

구성 파일의 목록에는 내보내기 파일 및 가져오기 위해 업로드한 모든 파일이 포함됩니다.

curl 명령은 다음과 같습니다.

```
curl -X GET --header 'Accept: application/json'
'https://10.89.5.38/api/fdm/최신/action/configfiles'
```

응답에는 항목 목록이 표시되며 각 항목은 구성 파일입니다. 예를 들어, 다음 목록에는 2개의 파일이 표시됩니다. 모든 파일의 **id**는 기본값입니다. ID를 무시하고 **diskFileName**을 대신 사용합니다.

```
{
  "items": [
    {
      "diskFileName": "export-config-2.zip",
      "dateModified": "2019-04-19 13:32:28Z",
      "sizeBytes": 10182,
      "id": "default",
      "type": "configimportexportfileinfo",
      "links": {
        "self": "https://10.89.5.38/api/fdm/최신/action/configfiles/default"
      }
    },
    {
      "diskFileName": "export-config-1.zip",
      "dateModified": "2019-04-19 13:14:56Z",
      "sizeBytes": 10083,
      "id": "default",
      "type": "configimportexportfileinfo",
      "links": {
        "self": "https://10.89.5.38/api/fdm/최신/action/configfiles/default"
```

```
      }
    }
  ],
```

단계 **2** diskFileName을 개체 ID로 사용하여 파일을 다운로드합니다.

curl 명령은 다음과 같습니다.

```
curl -X GET --header 'Accept: application/octet-stream'
'https://10.89.5.38/api/fdm/최신/action/downloadconfigfile/export-config-2.zip'
```

파일은 기본 다운로드 폴더에 다운로드됩니다. API Explorer에서 GET 메서드를 실행하고 브라우저가 다운로드 위치를 확인하도록 구성된 경우 파일을 저장하라는 메시지가 표시됩니다.

다운로드에 성공하면 반환 코드 200이 생성되며 응답 본문이 없습니다.

내보낸 구성 파일 수정

구성 파일을 다운로드한 후에는 해당 파일의 압축을 풀어 개체가 포함된 텍스트 파일을 열 수 있습니다. WordPad에서는 NotePad보다 읽기 쉬운 방식으로 콘텐츠를 포맷합니다. 설치되어 있는 다른 텍스트 편집기를 사용할 수도 있습니다. 고유한 구성 파일을 처음부터 생성할 수도 있지만, 파일 구조를 파악하려면 구성을 내보내야 합니다.

다음 주제에서는 텍스트 파일의 요구 사항에 대해 설명합니다.

최소 구성 파일 요구 사항

구성 파일에는 다음과 같은 최소 요소가 필요합니다.

- 파일의 개체를 [대괄호]로 묶습니다. 전체 파일은 표준 JSON 표시법을 사용하며 개체를 배열한 것입니다.
- 각 개체를 {중괄호}로 묶습니다.
- 쉼표를 사용하여 구성 파일의 개체를 구분합니다. 즉, 개체의 끝 중괄호 뒤에는 마지막 개체를 제외하고 쉼표가 와야 합니다.
- 파일의 첫 번째 개체는 메타데이터 개체여야 합니다. 올바른 개체 특성을 가져오는 가장 쉬운 방법은 원하는 모델의 디바이스에서 구성을 내보내는 것입니다. 예를 들어, 다음은 Secure Firewall Threat Defense Virtual 디바이스의 메타데이터 개체입니다. 디바이스를 가져오기 전에 구성 및 내보내기 유형을 수정할 수 있으며, 원하는 경우 generatedOn 특성을 삭제할 수 있습니다.

```
{"hardwareModel":"Cisco Firepower Threat Defense for VMWare",
 "type":"metadata",
 "configType":"FULL_CONFIG",
 "apiVersion":"최신",
 "generatedOn":"Fri Apr 19 13:32:28 UTC 2019",
 "exportType":"FULL_EXPORT",
 "softwareVersion":"6.5.0-10480"}
```

- 메타데이터 개체에서는 적절한 구성 유형(configType) 값을 지정해야 합니다.

- FULL_CONFIG - 이 텍스트 파일에는 전체 디바이스 구성이 포함됩니다.
- DELTA_CONFIG - 이 텍스트 파일에는 일부 개체만 포함하는 부분 구성이 포함됩니다.

- exportType은 FULL_EXPORT, PARTIAL_EXPORT, PENDING_CHANGE_EXPORT 중 하나입니다.
- 전체 구성 가져오기를 수행 중인 경우, 메타데이터 개체에서는 hardwareModel, softwareVersion, apiVersion의 특성을 지정해야 합니다.
- 개체를 하나 또는 여러 줄에 작성할 수 있지만, 개체의 특성 사이에는 빈 줄 또는 주석 줄을 넣지 마십시오. 파일에서는 주석이 허용되지 않습니다.
- 개체는 다른 개체에서 참조하는 개체가 먼저 정의되는 종속성 순서로 내보내기되지만, 가져오기 구성 파일에서 해당 순서를 유지할 필요는 없습니다. 시스템에서는 개체 이름 및 ID가 종속 개체 간에 올바르게 확인되는 것으로 간주하여 가져오기 중에 관계를 자동으로 확인합니다.

ID 래퍼 개체의 기본 구조

구성 파일에서는 ID 래퍼 개체를 사용하여 내보내거나 가져올 수 있는 ConfigEntity 또는 ManagementEntity 개체를 정의합니다. ID 래퍼 개체의 기본 구조는 다음과 같습니다.

```
{
    "type" : "identitywrapper",
    "data" : {},
    "parentName" : "container-name",
    "oldName" : "old-object-name",
    "action" : "EDIT", //Enum values: CREATE, EDIT or DELETE
    "index" : integer,
  }
```

개체에는 다음 특성이 포함됩니다.

- **type** - 항상 **identitywrapper**입니다.
- **data** - 네트워크 개체, 액세스 제어 규칙 등의 구성에서 개체를 정의하는 특성-값 쌍의 모음입니다. 이 모음에 필요한 특성은 특정 개체 유형과 수행 중인 작업을 위한 모델에 따라 달라집니다. 특성-값 쌍을 {중괄호}로 묶습니다. 데이터 배열 내의 특성을 쉼표로 구분합니다.
- **parentName** - (필요시) 제한된 수의 개체는 ContainedObjects이며 이를 포함하는 개체와 관계가 있습니다. 예로는 액세스 규칙, 수동 NAT 규칙, 하위 인터페이스가 있습니다. 이러한 항목의 경우 parentName에서은 포함하는 개체(상위 항목)의 이름을 지정합니다. 포함된 개체에 이 특성을 지정하고, 포함되지 않은 개체에는 이 특성을 지정하지 마십시오. 이러한 개체에 대한 인덱스를 지정해야 할 수도 있습니다.

  상위 항목이 단일 개체(즉, 둘 이상 생성할 수 없음, 예: AccessPolicy)이며 시스템에서 참조를 확인할 수 있는 경우 실제로 이 특성을 생략할 수 있습니다.
- **oldName** - (필요시) 기존 개체의 이름을 바꾸는 경우, 이 특성의 이전 이름 및 데이터 특성의 **name** 특성에 새 이름을 지정할 수 있습니다. 이 특성을 사용하려면 action이 EDIT여야 합니다.

- **action** - 정의된 개체와 관련하여 수행할 작업입니다. 전체 내보내기에서는 작업이 항상 **CREATE** 입니다. 보류 중인 변경 또는 부분 내보내기의 경우 다른 작업은 **EDIT** 또는 **DELETE**일 수 있습니다.

  가져오기를 위해 파일을 수정할 때 원하는 작업을 지정합니다. CREATE를 지정했지만 개체가 이미 존재하는 경우, 작업은 EDIT로 변경됩니다. 개체가 없는 경우, EDIT가 CREATE로 변경됩니다. DELETE 작업은 변경되지 않습니다. 개체 참조는 개체 유형 및 이름, 개체 유형 및 이전 이름, 또는 개체 유형 및 상위 이름에 따라 확인됩니다.

  - CREATE - 이는 새 개체입니다. 개체를 게시할 때 필요한 데이터 특성을 지정해야 합니다. **name**이 지정된 유형의 기존 개체와 일치하는 경우, 작업이 EDIT로 자동으로 변경됩니다.

    새 개체를 생성하고 다른 개체에서 해당 개체를 참조하는 경우(예: 네트워크 개체를 정의한 다음, 액세스 규칙에서 이를 사용하는 경우)에는 참조에서 개체 **name**이 정확해야 합니다.

  - EDIT - 개체를 업데이트하는 중입니다. version 및 id를 제외하고 개체를 배치할 때 필요한 데이터 특성을 지정해야 합니다. 이름 및 개체 유형은 업데이트할 개체를 결정하는 데 사용되며 버전 특성은 항상 무시됩니다.

  - DELETE - 개체를 삭제하는 중입니다. 개체 데이터에서 **type** 및 **name** 특성을 지정해야 합니다.

- **index** - (선택 사항, 정수) 액세스 제어 및 수동 NAT 규칙과 같이 순서가 지정된 목록의 일부인 개체의 경우, 정책에서 개체의 위치입니다. 새 규칙을 생성하고 인덱스 값을 지정하지 않은 경우, 규칙은 마지막 규칙으로 정책 끝에 추가됩니다. 규칙을 수정하는 중인 경우, 규칙의 기존 위치가 유지됩니다.

예: 다른 디바이스로 가져오기 위해 네트워크 개체 수정

각 개체는 다음과 같이 구성되며, 이는 syslog 서버의 IP 주소를 정의하는 네트워크 호스트 개체입니다.

```
{"type":"identitywrapper",
 "action":"CREATE",
 "data":{
    "version":"1fxdbtbyg4ex6",
    "name":"syslog-host",
    "subType":"HOST",
    "value":"10.100.10.10",
    "isSystemDefined":false,
    "dnsResolution":"IPV4_AND_IPV6",
    "id":"2cd0ea03-62a7-11e9-8b8d-dbf377c781d8",
    "type":"networkobject"}}
```

디바이스에서 이 개체를 내보냈으며 개체를 다른 디바이스로 가져오고 싶지만 새 디바이스가 다른 주소(192.168.5.15)에서 syslog 서버를 사용해야 한다고 가정해 보겠습니다. 새 개체를 생성할 예정이므로 데이터 특성에서 **version** 및 **id** 특성을 제거합니다. **isSystemDefined**(기본값이 false임) 및 **dnsResolution**(FQDN 개체만 해당)도 제거할 수 있습니다. 그 결과 생성되는 새 개체는 다음과 같이 표시됩니다.

```
{"type":"identitywrapper",
 "action":"CREATE",
```

```
 "data":{
    "name":"syslog-host",
    "subType":"HOST",
    "value":"192.168.5.15",
    "type":"networkobject"}}
```

파일 상단에서 메타데이터 개체를 유지(또는 추가)해야 합니다. 파일 콘텐츠를 더 쉽게 스캔하고 확인할 수 있도록 줄 반환을 추가할 수도 있습니다. 따라서 전체 구성 파일은 다음과 같이 표시됩니다.

```
[
{"hardwareModel":"Cisco Firepower Threat Defense for VMWare",
 "type":"metadata",
 "configType":"DELTA_CONFIG",
 "apiVersion":"최신",
 "exportType":"PARTIAL_EXPORT",
 "softwareVersion":"6.5.0-10465"}
,
{"type":"identitywrapper",
 "action":"CREATE",
 "data":{
    "name":"syslog-host",
    "subType":"HOST",
    "value":"192.168.5.15",
    "type":"networkobject"}}
]
```

가져오기 파일 업로드

구성 파일을 디바이스에 가져오려면 먼저 파일을 디바이스에 업로드해야 합니다. zip 또는 텍스트 파일 중 하나를 업로드할 수 있습니다. zip 파일을 사용하는 경우 AnyConnect 패키지 및 클라이언트 프로필을 포함할 수 있습니다.

POST /action/uploadconfigfile 리소스를 사용하여 파일을 업로드합니다. 이름의 최대 길이는 60자입니다.

- API Explorer에서 이 메서드를 사용하는 경우, **fileToUpload** 특성 옆에 있는 **Choose File(**파일 선택**)** 버튼을 클릭하여 워크스테이션 드라이브에서 파일을 선택합니다.
- 고유한 프로그램에서 이 메서드를 사용 중인 경우, 요청 페이로드는 파일 이름 필드와 함께 단일 파일 항목을 포함해야 합니다. 파일 이름 확장명은 .txt 또는 .zip이어야 하며, 실제 파일 콘텐츠 형식은 파일 확장명과 일치해야 합니다.

curl 명령은 다음과 같이 표시됩니다.

```
curl -F 'fileToUpload=@./import-1.txt'
'https://10.89.5.38/api/fdm/최신/action/uploadconfigfile'
```

전송에 성공하면 반환 코드 200 및 다음과 유사한 응답 본문이 표시됩니다. 여기에는 위협 방어 시스템의 파일 이름(**diskFileName**)이 표시되며, 이는 가져오기 작업에 필요합니다.

```
{
  "diskFileName": "import-1.txt",
  "dateModified": "2019-04-22 10:18:12Z",
  "sizeBytes": 267,
  "id": "default",
```

```
  "type": "configimportexportfileinfo",
  "links": {
    "self": "https://10.89.5.38/api/fdm/최신/action/uploadconfigfile/default"
  }
}
```

구성 가져오기 및 작업 상태 확인

위협 방어 시스템에 구성 파일을 업로드한 후에는 구성 파일에 정의된 개체를 위협 방어 구성에 가져올 수 있습니다. POST /action/configimport 메서드를 사용합니다.

개체를 가져올 때, 구성 파일이 아닌 import 명령으로 개체를 직접 정의하는 옵션을 사용할 수도 있습니다. 그러나 한두 개의 네트워크 개체 등 적은 수의 변경 사항을 가져오는 경우에만 개체를 직접 정의해야 합니다.

프로시저

단계 **1** 가져오기 작업용 JSON 개체 본문을 생성합니다.

이 호출에 사용할 JSON 개체의 예는 다음과 같습니다.

```
{
  "diskFileName": "string",
  "encryptionKey": "*********",
  "preserveConfigFile": true,
  "autoDeploy": true,
  "allowPendingChange": true,
  "excludeEntities": [
     "string"
     ]",
  "inputEntities": [
    {
      "action": "CREATE",
      "oldName": "string",
      "parentId": "string",
      "parentName": "string",
      "index": 0,
      "data": {
        "version": "string",
        "id": "string",
        "type": "identity"
      },
      "id": "string",
      "type": "IdEntityWrapper"
    }
  ],
  "jobName": "string",
  "type": "scheduleconfigimport"
}
```

특성은 다음과 같습니다.

- **diskFileName** - 가져올 구성 zip 또는 txt 파일의 이름입니다.
- **encryptionKey** - zip 파일을 암호화하는 데 사용된 키입니다(있는 경우). 구성 파일이 암호화되지 않은 경우 키를 지정하지 마십시오.

- **preserveConfigFile** - (선택 사항) 가져오기 작업을 성공적으로 완료한 후에 가져온 구성 파일의 복사본을 위협 방어 디스크에 보관할지 여부입니다. 파일을 보관하려면 true를 지정하고 위협 방어 디스크에서 파일을 삭제하려면 false를 지정합니다. 기본값은 false입니다.
- **autoDeploy** - (선택 사항) 가져오기에 성공한 경우 구축 작업을 자동으로 시작할지 여부입니다. 가져온 개체는 보류 중인 변경 사항에 해당하며 변경 사항을 성공적으로 구축할 때까지 활성화되지 않습니다. 구축 작업을 자동으로 시작하려면 true를 지정합니다. false를 지정하는 경우 변경 사항을 수동으로 구축해야 합니다. 기본값은 false입니다.
- **allowPendingChange** - (선택 사항) 기존의 보류 중인 변경 사항이 있는 경우 가져오기 작업을 시작할지 여부입니다. 이 특성을 true로 설정하고 autoDeploy를 true로 설정하면 자동 구축 작업에 기존 및 가져온 변경 사항 등 모든 변경 사항이 포함됩니다. 이 특성을 false로 설정하면 보류 중인 변경 사항이 있는 경우 가져오기 작업이 실행되지 않습니다. 기본값은 false입니다.
- **excludeEntities** - (선택 사항) 가져오지 않아야 하는 개체를 식별하는 개체 일치 문자열의 목록입니다. 가져오기 파일에 가져오지 않을 항목이 포함된 경우에만(즉, 업로드한 파일에서 삭제하지 않기로 결정한 경우) 이 특성을 지정해야 합니다. 이 목록의 각 항목은 "**id=***uuid-value*", "**type=***object-type*" 또는 "**name=***object-name*"rhk 같은 패턴을 지닙니다. 이러한 패턴 중 하나와 일치하는 입력 개체는 가져오기에서 제외됩니다.

  type은 리프 엔터티(예: networkobject)이거나 리프 유형 집합의 별칭일 수 있습니다. 몇 가지 일반적인 유형의 별칭으로는 네트워크(NetworkObject 및 NetworkObjectGroup), 포트(모든 TCP/UDP/ICMP 포트, 프로토콜 및 그룹 유형), url(URL 개체 및 그룹), ikepolicy(IKE V1/V2 정책), ikeproposal(Ike V1/V2 제안), identitysource(모든 id 소스), 인증서(모든 인증서 유형), 개체(device manager에서 개체 페이지에 나열되는 모든 개체/그룹 유형), 인터페이스(모든 네트워크 인터페이스, s2svpn(모든 Site-to-Site VPN 관련 유형), ravpn(모든 RA VPN 관련 유형), vpn(s2svpn 및 ravpn 둘 다)이 있습니다.

  예를 들어, 모든 네트워크 개체와 myobj 및 UUID라는 이름으로 식별된 두 개의 다른 개체를 가져오기에서 제외하려면 다음을 지정합니다.

  ```
  "excludeEntities": [
     "type=networkobject",
     "name=myobj",
     "id=acc2e3cd-8c70-11e9-930a-1f12ee87b286"
     ]",
  ```

- **inputEntities** - 가져올 개체 수가 적은 경우, 구성 파일이 아닌 inputEntities 개체 목록에서 정의할 수 있습니다. 이 특성을 사용하려면 diskFileName 특성을 포함할 수 없으며 이 특성을 null로 설정해야 합니다.
- **jobName** - (선택 사항) 내보내기 작업의 이름입니다. 작업의 이름을 지정하면 작업 상태를 검색할 때 더 쉽게 찾을 수 있습니다.
- **type** - 작업 유형이며 항상 **scheduleconfigimport**입니다.

예제**:**

다음 예에서는 import-1.txt라는 이름의 구성 파일을 가져옵니다.

```
{
  "diskFileName": "import-2.txt",
```

```
  "preserveConfigFile": true,
  "autoDeploy": true,
  "allowPendingChange": true,
  "type": "scheduleconfigimport"
}
```

단계 **2** 개체를 게시합니다.

예를 들어 curl 명령은 다음과 같이 표시됩니다.

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json'
 -d '{ \
   "diskFileName": "import-2.txt", \
   "preserveConfigFile": true, \
   "autoDeploy": true, \
   "allowPendingChange": true, \
   "type": "scheduleconfigimport" \
 }' 'https://10.89.5.38/api/fdm/최신/action/configimport'
```

단계 **3** 응답을 확인합니다.

응답 코드 200이 표시되어야 합니다. 최소 JSON 개체를 게시한 경우, 성공적인 응답 본문은 다음과 같이 표시됩니다. 암호화 키를 지정하는 경우 응답에서 해당 키가 마스킹됩니다.

```
{
  "version": null,
  "scheduleType": "IMMEDIATE",
  "user": "admin",
  "forceOperation": false,
  "jobHistoryUuid": "7e360139-6725-11e9-abb5-078014531401",
  "ipAddress": "10.24.127.37",
  "diskFileName": "import-2.txt",
  "encryptionKey": null,
  "preserveConfigFile": true,
  "autoDeploy": true,
  "allowPendingChange": true,
  "jobName": "Config Import",
  "id": "7e2b52d8-6725-11e9-abb5-5dec35337506",
  "type": "scheduleconfigimport",
  "links": {
    "self": "https://10.89.5.38/api/fdm/최신
/action/configimport/7e2b52d8-6725-11e9-abb5-5dec35337506"
  }
}
```

단계 **4** GET /jobs/configimportstatus를 사용하여 가져오기 작업의 상태를 확인합니다.

또는 GET /jobs/configimportstatus/{objId}를 사용하여 한 개의 가져오기 작업의 상태를 가져올 수 있습니다. objId의 경우 응답 본문의 jobHistoryUuid 값을 POST /action/configimport 호출에 사용합니다.

curl 명령은 다음과 같이 표시됩니다.

```
curl -X GET --header 'Accept: application/json'
'https://10.89.5.38/api/fdm/최신/jobs/configimportstatus'
```

성공적인 가져오기에 대한 응답 본문은 다음과 같이 표시될 수 있습니다. 가져오기에 실패하면 형식 지정 또는 콘텐츠 오류를 수정하기 위해 파일을 수정하고 다시 시도해야 할 수 있습니다.

```
{
```

```
      "version": "pcgccfnk4hmiz",
      "jobName": "Config Import",
      "jobDescription": null,
      "user": "admin",
      "startDateTime": "2019-04-25 06:43:54Z",
      "endDateTime": "2019-04-25 06:44:01Z",
      "status": "SUCCESS",
      "statusMessage": "The configuration was imported successfully",
      "scheduleUuid": "7e2b52d8-6725-11e9-abb5-5dec35337506",
      "diskFileName": "import-2.txt",
      "messages": [],
      "preserveConfigFile": true,
      "autoDeploy": true,
      "allowPendingChange": true,
      "id": "7e360139-6725-11e9-abb5-078014531401",
      "type": "configimportjobstatus",
      "links": {
        "self": "https://10.89.5.38/api/fdm/최신
/jobs/configimportstatus/7e360139-6725-11e9-abb5-078014531401"
  }
}
```


다음에 수행할 작업

autoDeploy를 false로 설정한 경우, 가져온 변경 사항을 통합하려면 구축 작업을 실행해야 합니다. POST /operational/deploy 메서드를 사용합니다. 이 메서드를 true로 설정한 경우 구성이 성공적으로 구축되어야 합니다. device manager 또는 API(GET /operational/auditevents)에서 감사 로그를 확인할 수 있으며, 구축 작업의 이름은 "구성 후 가져오기 구축"으로 지정되어 있습니다.

참고 일부 기능의 경우 특정 라이선스가 필요합니다. 예를 들어, 디바이스에는 원격 액세스 VPN 기능에 대한 라이선스가 있어야 합니다. 그러나 가져오기 프로세스에서는 라이선스를 검증하지 않습니다. 따라서 라이선스 제어 기능에 대한 개체를 필수 라이선스가 없는 디바이스로 가져오는 경우, 구축 작업에 실패합니다. 이 문제가 발생하면 디바이스에 필수 라이선스를 할당하거나 개체를 삭제합니다.

필요 없는 가져오기/내보내기 파일 삭제

구성 파일이 더 이상 필요하지 않은 경우, 내보내기 작업을 통해 생성된 파일 또는 구성 가져오기를 위해 업로드한 파일을 삭제할 수 있습니다.

파일 이름을 objId 값으로 사용하는 DELETE /action/configfiles/{objId} 메서드를 사용합니다.

예를 들어, export-config-2.zip이라는 이름의 파일을 삭제하려는 경우 curl 명령은 다음과 같습니다.

```
curl -X DELETE --header 'Accept: application/json'
'https://10.89.5.38/api/fdm/최신/action/configfiles/export-config-2.zip'
```

성공적인 결과는 응답 본문이 없는 204 반환 코드입니다.

GET /action/configfiles를 사용하여 파일이 삭제되었음을 확인할 수 있습니다.