配置导入/导出

版本要求：要使用配置导入/导出，必须运行 威胁防御版本 6.5(0) 或更高版本，以及 威胁防御 REST API v4 或更高版本。

可以从 设备管理器托管的设备导出配置，并将其导入至同一设备或另一个兼容设备。例如，可以使用配置导入/导出在多个类似设备之间复制基线配置，然后在每台设备上使用 设备管理器 配置各设备的独有特征。

关于配置导入/导出

在本地管理 威胁防御设备时，您可借助 设备管理器FDM 或 CDO 使用 威胁防御 API 导出设备配置。此方法不能与 Cisco Secure Firewall Management Center 管理的设备配合使用。

导出配置后，系统创建 zip 文件。然后，可以将 zip 文件下载至工作站。配置本身表示为在 JSON 格式文本文件中使用属性-值对定义的对象。在将文件导回至同一设备或其他设备前，可以对其进行编辑。

因此，可以使用导出文件创建可部署到网络中其他设备的模板。

导入对象时，还可以选择直接在导入命令中定义对象，而不是在配置文件中定义对象。但是，仅在导入少量更改的情况下，才应直接定义对象。

以下主题介绍有关配置导入/导出的更多信息。

导出文件中包含的内容

导出时，要指定导出文件中所含的配置。完整导出包括导出 zip 文件中的所有内容。根据所选的待导出内容，导出 zip 文件可能包括以下内容：

- 定义各已配置对象的属性-值对。所有可配置项均建模为对象，而不仅仅是 设备管理器 中称为“对象”的那些对象。

- 如果配置了远程接入 VPN，则包括 AnyConnect 软件包和任何其他引用文件，如客户端配置文件 XML 文件、DAP XML 文件和 Hostscan 软件包。
- 如果已配置自定义文件策略，则包括任何引用的干净列表或自定义检测列表。

比较导入/导出和备份/还原

配置导入/导出与备份/恢复不同。

- 备份/恢复用于灾难恢复。仅当设备是同一型号，且其运行的软件版本与进行备份的设备相同时，才能将备份恢复至设备。首先，这是为了将“最后正确的”配置恢复至同一设备，或将配置恢复至替换设备。
- 导入/导出用于保留全部或部分配置。可以使用导出文件在重新映像设备后将配置恢复至设备。或者，也可以使用导出文件作为模板，编辑其内容再将其导入至其他设备。通过导入/导出，可以快速地将新设备配置为特定基线配置，以便更快地将其部署至网络中。在相应限制范围内，甚至可以将文件导入至不同的设备型号，例如从 Firepower 2120 导入到 2130。如果导入文件仅包含所有设备型号上支持的对象，则导入的限制应非常少。其中一个限制是设备需要使用与导出文件相同的 API 版本。

导入/导出策略

以下是可以使用导入/导出的一些方法。

- **创建用于新设备的模板**。将您的型号设备配置为所需基线，然后导出完整配置。随后，可以将该配置导入新设备，然后使用 设备管理器 或 威胁防御 API 进行所需的任何修改。还可以在导入之前编辑模板，以进行上述修改，例如修改各接口的 IP 地址。请注意，完整导出包括 ManagementIP 对象 (type=managementip)；假设您已在目标设备上配置管理地址和网关，则在为新设备创建模板时，应从导出文件中删除此对象，否则将覆盖管理寻址信息。
- **将配置更改从一台设备部署至其他类似设备**。例如，在编辑设备 A 的配置时，会创建一些新的网络对象和访问控制规则。然后，可以导出待处理更改，并将这些更改导入设备 B。在两台设备上部署配置后，它们运行相同的新规则。
- **重新映像系统后重新应用配置**。重新映像设备会擦除配置。如果首先导出完整配置，则可以在完成重新映像后将其导入。
- **应用有针对性的配置**。由于可以编辑甚至手动创建导出文件，因此可以删除除要导入其他设备中的对象以外的所有对象。例如，可以创建包含一组网络对象的配置文件，并用该文件将相同的网络对象组导入至您的所有 威胁防御设备中。

配置导入/导出准则

- 导出作业期间，系统在配置数据库上保持写锁定。作业完成之前，无法使用 API 或 设备管理器 进行配置更改。但是，可以在导出作业期间查看 设备管理器 中的配置或使用 API 中的 GET 调用。
- 导入作业期间，系统在配置数据库上保持读写锁定。作业完成之前，无法使用 API 或 设备管理器 查看配置或对其进行更改。
- 导入配置会添加至现有配置。无法擦除设备配置，并将其替换为导入的配置。如果需要在导入前重置设备配置，则可转至设备 CLI 并发出 **configure manager delete** 命令，然后发出 **configure manager local** 命令。系统将仅保留管理接口配置。
- 仅当设备运行的 API 版本与文件中包含的元数据对象内 apiVersion 属性的定义相同时，才能将文件导入设备。

导入和导出配置

导入/导出过程始于从本地管理的设备导出配置。然后，可以下载导出文件，并根据需要进行编辑，然后再将其上传至同一设备或兼容设备。以下主题介绍各步骤。

导出配置

使用 POST /action/configexport 方法创建和开始配置导出作业。

过程

步骤 1 创建用于导出作业的 JSON 对象正文。

以下是要与此调用结合使用的 JSON 对象示例。

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

属性包括：

- **diskFileName** -（可选。）导出 zip 文件的名称。如果不指定名称，系统会为您生成一个名称。即使指定名称，系统也可能在名称之后附加某些字符以确保唯一性。名称的最大长度为 60 个字符。
- **encryptionKey** -（可选。) zip 文件的加密密钥。如果不想加密文件，请忽略此字段并指定 "doNotEncrypt": true。如果指定密钥，则在将 zip 文件下载至工作站后，需要使用该密钥打开 zip 文件。请注意，导出的配置文件以明文形式公开密钥、密码和其他敏感数据（否则将无法导入），因此您可能希望应用加密密钥来保护敏感数据。系统使用 AES 256 加密。
- **doNotEncrypt** -（可选。）导出文件应该加密 (false) 还是不加密 (true)。默认值为 false，这意味着必须指定非空的 encryptionKey 属性。如果指定 true，则将忽略 encryptionKey 属性。
- **configExportType** - 以下枚举值之一：
  - **FULL_EXPORT** - 在导出文件中包括整个配置。这是默认值。
  - **PARTIAL_EXPORT** - 仅包含在 entityIds 列表中标识的对象及其后代对象。对于不可导出的对象，即使指定其身份，也无法将其包括在内。所有用户定义的对象均可导出。
  - **PENDING_CHANGE_EXPORT** - 仅包括尚未部署的对象，即待处理更改。
- **deployedObjectsOnly** -（可选。）是否仅在对象已部署时才将其包含在导出文件中。也就是说，不包括待处理更改。对于 PENDING_CHANGE_EXPORT 作业，忽略此属性，因为这些作业仅包括未部署对象。默认值为 false，表示导出中包含所有待处理更改。指定 true 以排除待处理更改。
- **entityIds** - 一组起始点对象的身份列表，其中对象以逗号分隔并括在 [方括号] 中。PARTIAL_EXPORT 作业需要此列表。此列表中的各项均可以是 UUID 值或与 "**id=***uuid-value*"、"**type=***object-type*" 或 "**name=***object-name*" 等模式匹配的属性值对。例如，**"type=networkobject"**。

  type 可以是叶实体（例如 networkobject），也可以是一组叶类型的别名。一些典型的类型别名包括：network（NetworkObject 和 NetworkObjectGroup）、port（所有 TCP/UDP/ICMP 端口、协议和组类型）、url（URL 对象和组）、ikepolicy（IKE V1/V2 策略）、ikeproposal（Ike V1/V2 提议）、identitysource（所有身份源），certificate（所有证书类型）、object（将在“对象”(Objects) 页面上的 设备管理器 中列出的所有对象/组类型）、interface（所有网络接口）、s2svpn（所有站点间 VPN 相关类型）、ravpn（所有 RA VPN 相关类型）和 vpn（s2svpn 和 ravpn）。

  所有这些对象及其传出引用后代将包含在 PARTIAL_EXPORT 输出文件中。请注意，所有不可导出对象都将从输出中排除，即使您指定其身份。使用相应资源类型的 GET 方法获取目标对象的 UUID、类型或名称。

  例如，要导出所有网络对象以及名为 myaccessrule 的访问规则和由 UUID 标识的两个对象，可指定：

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

- **jobName** -（可选。）导出作业的名称。在检索作业状态时，给出作业名称可以更轻松地进行查找。
- **type** - 作业类型，始终为 **scheduleconfigexport**。

示例：

以下示例对文件 export-config-1 执行完全导出，并接受所有其他属性的默认值：

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

步骤 2 发布对象。

例如，curl 命令会如下所示：

```
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/最新/action/configexport'
```

步骤 3 验证响应。

您应获得的响应代码为200。如果发布了最小的JSON对象，则成功的响应正文将类似于以下内容。如果指定加密密钥，则会在响应中屏蔽该密钥。

```
{
  "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/最新
/action/configexport/c79be920-629a-11e9-8b8d-85231be77de0"
  }
}
```

检查导出作业的状态

导出作业需要一些时间才能完成。配置越大，作业所需的时间就越多。检查作业状态，确保其成功完成，然后再尝试下载文件。

获取状态的最简单方法是使用 GET/jobs/configexportstatus。例如，curl 命令会如下所示：

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

已成功完成的作业将返回类似于以下内容的状态。

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

也可以使用 GET /jobs/configexportstatus/{objId} 方法检索特定作业的状态。您会从响应对象的 **id** 字段中获取对象 ID。

下载导出文件

导出作业完成后，系统会将导出文件写入系统磁盘，并将其称为配置文件。可以使用 GET /action/downloadconfigfile/{objId} 方法将此导出文件下载至工作站。要获取可用文件的列表，请使用 GET /action/configfiles 方法。

注释 对于 GET /action/downloadconfigfile/{objId}，通常将文件名指定为对象 ID。或者，也可以指定与该文件相关联的 ConfigExportStatus 对象的 ID。

过程

步骤 1 获取磁盘上的配置文件列表。

配置文件列表包括导出文件和上传用于导入的任何文件。

curl 命令类似于下文：

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

响应将显示项目列表，每个项目都是一个配置文件。例如，以下列表显示 2 个文件。请注意，所有文件的 **id** 均是默认值。忽略 ID，并用 **diskFileName** 代替它。

```
{
  "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/最新/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/最新/action/configfiles/default"
      }
    }
  ],
```

步骤 2 使用 diskFileName 作为对象 ID 下载文件。

curl 命令类似于下文：

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

文件将下载至默认下载文件夹。如果是从 API Explorer 发出 GET 方法，且浏览器配置为提示下载位置，系统会提示您保存文件。

下载成功后会出现 200 返回代码，但无响应正文。

编辑导出的配置文件

下载配置文件后，可以将其解压缩并打开包含这些对象的文本文件。WordPad 格式比 NotePad 格式更便于阅读。还可以使用您可能已安装的其他文本编辑器。甚至可以从头创建自己的配置文件，但需要导出配置以了解文件结构。

以下主题介绍文本文件的要求。

最低配置文件要求

配置文件必须具有以下最小元素：

- 将文件中的对象放在 [方括号] 内。整个文件使用标准 JSON 表示法，是一组对象。
- 将各对象包含在 {大括号} 中。
- 使用逗号分隔配置文件中的对象。也就是说，对象的右括号后面应跟一个逗号，但最后一个对象除外。
- 文件中的第一个对象必须是元数据对象。获取正确对象属性的最简单方法是从所需模型的设备中导出配置。例如，以下是来自 Cisco Secure Firewall Threat Defense Virtual 设备的元数据对象。导入设备之前，可以编辑配置和导出类型，如果需要，请删除 generatedOn 属性。

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

- 元数据对象必须指定适当的配置类型 (configType) 值。
  - FULL_CONFIG - 此文本文件包含完整设备配置。
  - DELTA_CONFIG - 此文本文件包含部分配置，甚至只是几个对象。
- ExportType 是以下类型之一：FULL_EXPORT、PARTIAL_EXPORT、PENDING_CHANGE_EXPORT。
- 如果正在进行完全配置导入，元数据对象必须指定以下属性：hardwareModel、softwareVersion、apiVersion。
- 可以在一行或多行写入对象，但不要在对象中的属性之间添加空行或注释行。文件中不允许使用注释。
- 尽管对象按依赖关系项顺序导出，即首先定义被其他对象引用的对象，但在导入配置文件中无需保留该顺序。系统将在导入期间自动解析关系，假定在相关对象之间会正确解析对象名称和 ID。

身份封装对象基本结构

配置文件使用身份封装对象定义任何可以导出或导入的 ConfigEntity 或 ManagementEntity 对象。以下是身份封装对象的基本结构：

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

```
  "index" : integer,
}
```

该对象包含以下属性：

- **type** - 始终是 **identitywrapper**。
- **data** - 这是从配置中定义对象的属性-值对的集合，例如网络对象、访问控制规则等。此集合所需的属性取决于特定对象类型的型号以及您正在执行的操作。将属性-值对括在 {大括号} 中。使用逗号分隔数据数组中的各个属性。
- **parentName** -（如有需要）。有限数量的对象是 ContainedObjects，与包含这些对象的对象之间有关系。示例包括访问规则、手动 NAT 规则和子接口。对于这些项目，parentName 指定包含其他对象的对象（即父对象）的名称。为被包含于其他对象内的对象指定该属性。对于未包含于其他对象内的对象，请勿指定该属性。您可能还需要指定这些对象的索引。

  如果父对象是单个对象（即，您无法创建多个对象），例如 AccessPolicy，且系统可以解析引用，则实际上可以忽略此属性。
- **oldName** -（如有需要）。如果要重命名现有对象，可以在此属性上指定旧名称，并在数据属性的**名称**属性中指定新名称。必须使用“编辑”操作才能使用此属性。
- **action** - 要对已定义对象执行的操作。在完整导出中，操作始终是**创建**。对于待处理的更改或部分导出，可能会执行**编辑**或**删除**其他操作。

  编辑导入文件时，请指定所需操作。请注意，如果指定“创建”，但对象已存在，则会将操作更改为“编辑”；如果对象不存在，则会将“编辑”更改为“创建”。“删除”操作不会更改。对象引用根据对象类型和名称、对象类型和旧名称或对象类型和父名称予以解析。

  - 创建 - 这是个新对象。您需要指定发布对象时所需的数据属性。请注意，如果**名称**与指定类型的现有对象匹配，则该操作会自动更改为“编辑”。

    请注意，如果创建新对象并从其他对象引用该对象（例如定义网络对象，然后在访问规则中使用该对象），则该引用中的对象**名称**必须正确。
  - 编辑 - 更新对象。您需要指定放置对象时所需的数据属性，但版本和 ID 除外。名称和对象类型用于确定要更新的对象，且版本属性始终被忽略。
  - 删除 - 删除对象。您必须在对象数据中指定**类型**和**名称**属性。
- **index** -（可选；整数。）对于属于有序列表一部分的对象，例如访问控制和手动 NAT 规则，该属性是指对象在策略中的位置。如果要创建新规则而不指定索引值，则该规则将添加至策略末尾作为最新规则。如果您正在编辑规则，则系统将保留该规则的现有位置。

示例：编辑网络对象以导入至其他设备

各对象的结构如下所示，这是一个定义系统日志服务器 IP 地址的网络主机对象：

```
{"type":"identitywrapper",
 "action":"CREATE",
 "data":{
    "version":"lfxdbtbyg4ex6",
    "name":"syslog-host",
```

```
    "subType":"HOST",
    "value":"10.100.10.10",
    "isSystemDefined":false,
    "dnsResolution":"IPV4_AND_IPV6",
    "id":"2cd0ea03-62a7-11e9-8b8d-dbf377c781d8",
    "type":"networkobject"}}
```

假设您从设备导出此对象，且想要将该对象导入其他设备，但新设备应使用位于不同地址192.168.5.15上的系统日志服务器。由于您要创建新对象，请从数据属性中删除**版本**和 **ID** 属性。还可以删除 **isSystemDefined**（默认值为 false）和 **dnsResolution**（仅适用于 FQDN 对象）。生成的新对象如下所示：

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

在该文件的顶部，需要保留（或添加）元数据对象。您还可以添加换行符，以便更容易地扫描和验证文件内容。因此，完整的配置文件可能如下所示：

```
[
{"hardwareModel":"Cisco Firepower Threat Defense for VMWare",
 "type":"metadata",
 "configType":"DELTA_CONFIG",
 "apiVersion":"最新",
 "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"}}
]
```

上传导入文件

必须先将文件上传至设备，然后才能将配置文件导入设备。可以上传 zip 或文本文件。如果使用 zip 文件，则可以包括 AnyConnect 软件包和客户端配置文件。

使用 POST/action/uploadconfigfile 资源上传文件。名称的最大长度为 60 个字符。

- 如果从 API Explorer 使用此方法，请点击 **fileToUpload** 属性旁的**选择文件 (Choose File)** 按钮，以从工作站驱动器选择文件。
- 如果您从自己的程序使用该方法，则请求负载必须包含带有文件名称字段的单个文件项。文件扩展名必须是 .txt 或 .zip，且实际的文件内容格式必须与文件扩展名一致。

curl 命令会如下所示：

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

如果传输成功，则会出现返回代码 200 和类似于以下内容的响应正文，其中显示导入作业所需的 威胁防御系统 (**diskFileName**) 上的文件名。

```
{
  "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/最新/action/uploadconfigfile/default"
  }
}
```

导入配置并检查作业状态

将配置文件上传至 威胁防御系统后，可以将配置文件中定义的对象导入到 威胁防御 配置中。使用 POST /action/configimport 方法。

导入对象时，还可以选择直接在导入命令中定义对象，而不是在配置文件中定义对象。但是，仅在导入少量更改的情况下（例如一两个网络对象），才应直接定义对象。

过程

步骤 1 创建用于导入作业的 JSON 对象正文。

以下是要与此调用结合使用的 JSON 对象示例。

```
{
  "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"
}
```

属性包括：

- **diskFileName** - 要导入的配置 zip 或 txt 文件的名称。
- **encryptionKey** - 用于加密 zip 文件的密钥（如有）。如果配置文件未加密，请勿指定密钥。
- **preserveConfigFile** -（可选。）在成功导入作业后，是否在 威胁防御 磁盘上保留导入的配置文件的副本。指定 true 以保留文件，指定 false 以使文件从 威胁防御 磁盘中删除。默认值为 false。
- **autoDeploy** -（可选。）导入成功时是否自动开始部署作业。导入的对象是待处理更改，在成功部署更改前，这些对象处于非活动状态。指定 true 以自动开始部署作业。如果指定 false，则必须手动部署更改。默认值为 false。
- **allowPendingChange** -（可选。）如有现有待处理更改，是否允许开始导入作业。如果将此属性设置为 true，并将 autoDeploy 设置为 true，则自动部署作业将包括已预先存在和已导入的所有更改。如果将此属性设置为 false，则如果存在待处理更改，导入作业将不会运行。默认值为 false。
- **excludeEntities** -（可选。）标识不应导入的对象的对象匹配字符串列表。仅当导入文件包含您不想导入的项目（即，您决定不从上传文件中删除这些项目）时，才需指定此属性。此列表中的各项目模式如下："**id**=*uuid-value*"、"**type**=*object-type*" 或 "**name**=*object-name*"。系统将从导入中删除与其中一个模式匹配的输入对象。

  type 可以是叶实体（例如 networkobject），也可以是一组叶类型的别名。一些典型的类型别名包括：network（NetworkObject 和 NetworkObjectGroup）、port（所有 TCP/UDP/ICMP 端口、协议和组类型）、url（URL 对象和组）、ikepolicy（IKE V1/V2 策略）、ikeproposal（Ike V1/V2 提议）、identitysource（所有身份源），certificate（所有证书类型）、object（将在“对象”(Objects) 页面上的 设备管理器 中列出的所有对象/组类型）、interface（所有网络接口）、s2svpn（所有站点间 VPN 相关类型）、ravpn（所有 RA VPN 相关类型）和 vpn（s2svpn 和 ravpn）。

  例如，要排除导入所有网络对象以及由名称 myobj 和 UUID 标识的其他两个对象，请指定以下定义：

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

- **inputEntities** - 如果要导入的对象数量很少，则可以在 inputEntities 对象列表中而不是配置文件中对其进行定义。要使用此属性，则不能包含 diskFileName 属性，否则必须将该属性设置为 null。
- **jobName** -（可选。）导出作业的名称。在检索作业状态时，给出作业名称可以更轻松地进行查找。
- **type** - 作业类型，始终为 **scheduleconfigimport**。

示例：

以下示例导入名为 import-1.txt 的配置文件：

```
{
  "diskFileName": "import-2.txt",
  "preserveConfigFile": true,
  "autoDeploy": true,
  "allowPendingChange": true,
  "type": "scheduleconfigimport"
}
```

步骤 2 发布对象。

例如，curl 命令会如下所示：

```
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/最新/action/configimport'
```

步骤 3 验证响应。

您应获得的响应代码为200。如果发布了最小的JSON对象，则成功的响应正文将类似于以下内容。如果指定加密密钥，则会在响应中屏蔽该密钥。

```
{
  "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/最新
/action/configimport/7e2b52d8-6725-11e9-abb5-5dec35337506"
  }
}
```

步骤 4 使用 GET /jobs/configimportstatus 检查导入作业的状态。

或者，也可以使用 GET/jobs/configimportstatus/{objId} 获取导入作业的状态。对于 objId，请使用对 POST /action/configimport 调用的响应正文中的 jobHistoryUuid 值。

curl 命令会如下所示：

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

成功导入的响应正文可能如下所示。如果导入失败，可能需要编辑文件以更正格式或内容错误，然后重试。

```
{
      "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/最新
/jobs/configimportstatus/7e360139-6725-11e9-abb5-078014531401"
  }
}
```

下一步做什么

如果将 autoDeploy 设置为 false，则需要运行部署作业以应用导入的更改。使用 POST /operational/deploy 方法。如果将其设置为 true，则应已成功部署设置。在 设备管理器 或 API (GET/operational/auditevents) 中，可以检查审核日志，并将部署作业命名为“后配置导入部署”(Post Configuration Import Deployment)。

注释 某些功能需要特定的许可证。例如，设备必须具有用于任何远程接入 VPN 功能的许可证。但是，导入过程不验证许可证。因此，如果将许可证控制功能的对象导入至无所需许可证的设备中，则部署作业将失败。如果遇到此问题，请将所需许可证分配给设备，或删除这些对象。

删除不需要的导入/导出文件

如果不再需要配置文件（由导出作业创建的配置文件或上传用于配置导入的配置文件），则可删除该配置文件。

在将文件名作为 objId 值的情况下，请使用 DELETE /action/configfiles/{objId} 方法。

例如，要删除名为 export-config-2.zip 的文件，curl 命令如下所示：

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

成功的结果是出现 204 返回代码，但无响应正文。

可以使用 GET /action/configfiles 确认文件是否已删除。

当地语言翻译版本说明

思科可能会在某些地方提供本内容的当地语言翻译版本。请注意，翻译版本仅供参考，如有任何不一致之处，以本内容的英文版本为准。