# Importation/exportation de la configuration

**Exigence relative à la version** : pour utiliser l'importation/exportation de la configuration, vous devez exécuter la version 6.5(0) ou supérieure de Défense contre les menaces, et la version v4 ou supérieure de l'API REST Défense contre les menaces.

Vous pouvez exporter la configuration d'un dispositif géré avec le gestionnaire d'appareil et l'importer dans le même dispositif ou dans un autre dispositif compatible. Par exemple, vous pouvez utiliser l'importation/exportation de la configuration pour reproduire une configuration de référence sur plusieurs dispositifs similaires, puis utiliser le gestionnaire d'appareil sur chaque dispositif pour configurer les caractéristiques uniques de chaque dispositif.



# À propos de l'importation et de l'exportation de la configuration

Lorsque vous gérez le dispositif Défense contre les menaces localement, avec le gestionnaire d'appareil ou par l'intermédiaire du Security Cloud Control, vous pouvez exporter la configuration du dispositif à l'aide de l'API Défense contre les menaces. Cette méthode ne fonctionne pas avec un dispositif géré par le Cisco Secure Firewall Management Center.

Lorsque vous exportez la configuration, le système crée un fichier zip. Vous pouvez ensuite télécharger le fichier zip sur votre ordinateur. La configuration elle-même est représentée sous forme d'objets définis à l'aide de paires attribut-valeur dans un fichier texte au format JSON. Vous pouvez modifier le fichier avant de le réimporter dans le même dispositif ou dans un autre.

Ainsi, vous pouvez utiliser un fichier d'exportation pour créer un modèle que vous pouvez déployer sur d'autres dispositifs de votre réseau.

Lorsque vous importez des objets, vous avez également la possibilité de définir les objets directement dans la commande d'importation plutôt que dans un fichier de configuration. Cependant, vous ne devriez définir directement des objets que dans les cas où vous importez un petit nombre de modifications.

Les rubriques suivantes expliquent plus en détail les importations/exportations de configuration.

# Ce qui est inclus dans le fichier d'exportation

Lorsque vous procédez à une exportation, vous devez préciser quelles configurations doivent être incluses dans le fichier d'exportation. Une exportation complète comprend tout ce qui se trouve dans le fichier d'exportation zip. En fonction de ce que vous choisissez d'exporter, le fichier d'exportation zip peut comprendre ce qui suit :

- Paires attribut-valeur qui définissent chaque objet configuré. Tous les éléments configurables sont modélisés en tant qu'objets, et pas seulement ceux qui sont appelés « objects » (objets) dans le gestionnaire d'appareil.
- Si vous avez configuré le VPN d'accès à distance, les paquets AnyConnect et tous les autres fichiers référencés, tels que les fichiers XML du profil client, le fichier XML DAP et les paquets Hostscan.
- Si vous avez configuré des stratégies de fichiers personnalisées, toute liste de nettoyage référencée ou toute liste de détection personnalisée.

# Comparer l'importation/exportation et la sauvegarde/restauration

L'importation/exportation de la configuration n'est pas identique à la sauvegarde/restauration.

- La sauvegarde/restauration sert à récupérer les sinistres. Vous pouvez restaurer une sauvegarde sur un dispositif uniquement si celui-ci est du même modèle et exécutant la même version de logiciel que le dispositif à partir duquel la sauvegarde a été effectuée. Principalement, il s'agit de récupérer la « dernière bonne » configuration sur le même dispositif ou de restaurer la configuration sur un dispositif de remplacement.
- Importer/Exporter permet de conserver tout ou partie d'une configuration. Vous pouvez utiliser un fichier d'exportation pour restaurer la configuration sur un dispositif après l'avoir réinitialisé. Sinon, vous pouvez utiliser le fichier d'exportation comme modèle et modifier le contenu avant de l'importer dans un autre dispositif. Importer/Exporter vous permet de configurer rapidement un nouveau dispositif jusqu'à un certain niveau de base, afin de le déployer plus rapidement dans votre réseau. Avec certaines limites, vous pouvez même importer un fichier sur différents modèles de dispositif, par exemple, d'un châssis Firepower 2120 à un 2130. Si le fichier d'importation ne comprend que des objets pris en charge sur tous les modèles de dispositifs, il ne devrait pas y avoir beaucoup de restrictions à l'importation. La seule restriction est que le dispositif doit utiliser la même version d'API utilisée pour le fichier d'exportation.

# Stratégies pour l'importation/exportation

Voici quelques façons d'utiliser l'importation/exportation.

- **Créez un modèle pour les nouveaux dispositifs.** Configurez votre dispositif de modèle avec la référence dont vous avez besoin, puis exportez la configuration complète. Vous pourrez ensuite importer cette configuration dans de nouveaux dispositifs, puis utiliser l'API gestionnaire d'appareil ou Défense contre les menaces pour apporter les modifications nécessaires. Vous pouvez également modifier le modèle avant l'importation pour apporter ces modifications, par exemple aux adresses IP de chaque interface. Notez que l'exportation complète comprend l'objet ManagementIP (type=managementip); en supposant que vous avez déjà configuré l'adresse de gestion et la passerelle sur le dispositif cible, vous devez supprimer cet objet du fichier d'exportation lorsque vous créez le modèle pour le nouveau dispositif, ou vous remplacerez les informations d'adressage de gestion.

- **Déployer les modifications de configuration d'un dispositif sur d'autres dispositifs similaires.** Par exemple, lors de la modification de la configuration du dispositif A, vous créez quelques nouveaux objets réseau et règles de contrôle d'accès. Vous pouvez ensuite exporter les modifications en cours et importer ces modifications dans le dispositif B. Après avoir déployé la configuration sur les deux dispositifs, ils exécutent les mêmes nouvelles règles.
- **Réappliquez la configuration après avoir recréé l'image du système.** Le fait de recréer l'image d'un dispositif efface la configuration. Si vous exportez la configuration complète pour la première fois, vous pouvez l'importer après avoir terminé la recréation de l'image.
- **Appliquer les configurations ciblées.** Étant donné que vous pouvez modifier ou même créer manuellement un fichier d'exportation, vous pouvez supprimer tous les objets, à l'exception de ceux que vous souhaitez importer dans un autre dispositif. Par exemple, vous pouvez créer un fichier de configuration qui contient un ensemble d'objets réseau et l'utiliser pour importer le même groupe d'objets réseau dans tous vos dispositifs Défense contre les menaces.

# Lignes directrices pour les importations/exportations de configuration

- Pendant une tâche d'exportation, le système s'assure que l'écriture soit verrouillée sur la base de données de configuration. Vous ne pouvez pas utiliser l'API ni le gestionnaire d'appareil, pour modifier la configuration tant que la tâche n'est pas terminée. Cependant, vous pouvez consulter la configuration dans le gestionnaire d'appareil ou utiliser des appels GET dans l'API pendant la tâche d'exportation.
- Pendant une tâche d'importation, le système s'assure que l'écriture et la lecture soient verrouillées sur la base de données de configuration. Vous ne pouvez pas utiliser l'API ni le gestionnaire d'appareil, pour consulter ou modifier la configuration tant que la tâche n'est pas terminée.
- La configuration importée est ajoutée à la configuration existante. Vous ne pouvez pas effacer la configuration du dispositif et la remplacer par la configuration importée. Si vous devez réinitialiser la configuration du dispositif avant l'importation, vous pouvez accéder à l'interface de ligne de commande du dispositif et exécuter la commande **configure manager delete**, suivie de la commande **configure manager local**. Seule la configuration de l'interface de gestion sera conservée.
- Vous pouvez importer un fichier dans un dispositif uniquement si celui-ci exécute la même version de l'API que celle définie dans l'attribut apiVersion (versionDeL'Api) dans l'objet metadata (métadonnées) contenu dans le fichier.
- La version de SRU doit être la même sur les dispositifs d'exportation et d'importation, sinon l'importation échouera.

# Importer et exporter des configurations

Le processus d'importation et d'exportation commence par l'exportation de la configuration à partir d'un dispositif géré localement. Vous pouvez ensuite télécharger le fichier d'exportation, et éventuellement le modifier, avant de le téléverser sur le même dispositif ou dans un dispositif compatible. Les rubriques suivantes expliquent chaque étape.

# Exporter la configuration

Utilisez la méthode POST /action/configexport pour créer et démarrer une tâche d'exportation de la configuration.

### Procédure

**Étape 1** Créez le corps de l'objet JSON pour la tâche d'exportation.

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

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

Les attributs sont :

- **diskFileName** (nomDuFichierDeDisque) : (facultatif.) Le nom du fichier zip d'exportation. Si vous ne spécifiez pas de nom, le système en générera un pour vous. Même si vous spécifiez un nom, le système peut ajouter des caractères au nom pour garantir l'unicité. Le nom a une longueur maximale de 60 caractères.
- **encryptedKey** (cléChiffrée) : (facultatif.) Une clé de chiffrement pour le fichier zip. Si vous ne voulez pas chiffrer le fichier, ignorez ce champ et spécifiez plutôt « "doNotEncrypt": true » (« nePasChiffrer » : vrai). Si vous spécifiez une clé, vous devrez l'utiliser pour ouvrir le fichier zip après l'avoir téléchargé sur votre ordinateur. Remarquez que le fichier de configuration exporté expose des clés secrètes, des mots de passe et d'autres données sensibles en texte clair (car, sinon, ils ne peuvent pas être importés), de sorte qu'il est probable que vous souhaitiez y appliquer une clé de chiffrement pour protéger les données sensibles. Le système utilise le chiffrement AES 256.
- **doNotEncrypt** (nePasChiffrer) : (facultatif.) Indique si le fichier d'exportation doit être chiffré (false [faux]) ou non (true [vrai]). La valeur par défaut est false (faux), ce qui signifie que vous devez spécifier un attribut encryptionKey (cléDeChiffrement) non vide. Si vous spécifiez true (vrai), l'attribut encryptionKey (cléDeChiffrement) est ignoré.
- **configExportType** (typeD'ExportationDeConfiguration) : l'une des valeurs enum (énumération) suivantes :
  - **FULL_EXPORT** (EXPORTATION_COMPLÈTE) : comprend toute la configuration dans le fichier d'exportation. Il s'agit du paramètre par défaut.
  - **PARTIAL_EXPORT** (EXPORTATION_PARTIELLE) : n'inclure que les objets et leurs objets descendants qui sont identifiés dans la liste entityIds (identifiantsD'Identité). Les objets non exportables ne sont pas inclus, même si vous spécifiez leurs identités. Tous les objets définis par l'utilisateur sont exportables.

  - **PENDING_CHANGE_EXPORT** (EXPORTATION_DES_MODIFICATIONS_EN_ATTENTE) : n'inclure que les objets qui n'ont pas encore été déployés, c'est-à-dire les modifications en attente.

- **DeployedObjectsOnly** (ObjetsDéployésSeulement) : (facultatif.) Indique si les objets doivent être inclus dans le fichier d'exportation uniquement s'ils ont été déployés. C'est-à-dire qu'il ne faut pas inclure les modifications en attente. Cet attribut est ignoré pour les tâches PENDING_CHANGE_EXPORT, car ces tâches n'incluent que les objets non déployés. La valeur par défaut est false (faux), ce qui signifie que tout changement en attente est inclus dans l'exportation. Spécifiez true (vrai) pour exclure les changements en attente.
- **entityIds** (identifiantsD'Entité) : une liste d'identités séparées par des virgules avec un ensemble d'objets de point de départ, entre [crochets]. La liste est nécessaire pour une tâche PARTIAL_EXPORT (EXPORTATION_PARTIELLE). Chaque élément de cette liste peut être une valeur UUID ou une paire attribut-valeur correspondant à des motifs tels que « **id=***uuid-value* », « **type**=*objet-type* » ou « **name**=*objet*-name ». Par exemple, **« type=networkobject »**

  Le **type** peut être soit une entité leaf, telle qu'un networkobject (objetréseau), ou un alias d'un ensemble de types leaf. Voici quelques alias de types typiques : network (NetworkObject et NetworkObjectGroup), port (tous les types de port, de protocole et de groupe TCP/UDP/ICMP), url (objets et groupes URL), ikepolicy (politiques IKE V1/V2), ikeproposal (propositions Ike V1/V2), identitysource (toutes les sources d'identité), certificate (tous les types de certificats), object (tous les types d'objets/groupes qui seraient répertoriés dans le gestionnaire d'appareil sur la page Objects), interface (toutes les interfaces réseau, s2svpn (tous les types de VPN site à site), ravpn (tous les types de VPN RA), vpn (s2svpn et ravpn).

  Tous ces objets et leurs descendants référentiels sortants seront inclus dans le fichier de sortie PARTIAL_EXPORT. Remarquez que tous les objets non exportables seront exclus de la sortie même si vous spécifiez leurs identités. Utilisez la méthode GET pour les types de ressources appropriés afin d'obtenir les UUID, les types ou les noms des objets cibles.

  Par exemple, pour exporter tous les objets réseau, ainsi qu'une règle d'accès nommée myaccessrule et deux objets identifiés par UUID, vous pouvez spécifier :

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

- **jobName** (nomDeLaTâche) : (facultatif.) Un nom pour la tâche d'exportation. Donner un nom à la tâche d'exportation permet de la retrouver plus facilement lorsque vous récupérez le statut de la tâche.
- **type** (type) : le type de tâche, qui est toujours **scheduleconfigexport** (exportationdelaconfigurationducalendrier).

**Exemple :**

Dans l'exemple suivant, on effectue une exportation complète vers le fichier export-config-1 et on accepte les valeurs par défaut pour tous les autres attributs :

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

**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 '{ \
   "configExportType": "FULL_EXPORT", \
   "type": "scheduleconfigexport" \
 }' 'https://10.89.5.38/api/fdm/dernière version/action/configexport'
```

**Étape 3** Vérifiez la réponse.

Vous devriez obtenir un code réponse de 200. Si vous publiez l'objet JSON minimum, le corps de la réponse ressemblera à ce qui suit : Si vous spécifiez une clé de chiffrement, elle sera masquée dans la réponse.


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


# Vérifier l'état de la tâche d'exportation

La réalisation d'une tâche d'exportation prend un certain temps. Plus la configuration est volumineuse, plus de temps la tâche prendra. Vérifiez le statut de la tâche pour vous assurer qu'elle se réalise complètement avant d'essayer de télécharger le fichier.

La manière la plus simple de récupérer l'état est d'utiliser GET /jobs/configexportstatus. Par exemple, la commande curl ressemblerait à ce qui suit :

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

Une tâche terminée avec succès renverra un état semblable à ce qui suit.


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

Vous pouvez également utiliser la méthode GET /jobs/configexportstatus/{objId} pour récupérer l'état d'une tâche spécifique. Vous obtiendrez l'identifiant d'objet dans le champ **id** (identifiant) de l'objet de réponse.

# Télécharger le fichier d'exportation

Lorsqu'une tâche d'exportation est terminée, le fichier d'exportation est écrit sur le disque système et est appelé fichier de configuration. Vous pouvez télécharger ce fichier d'exportation sur votre ordinateur à l'aide de la méthode GET /action/downloadconfigfile/{objId}. Pour obtenir une liste des fichiers disponibles, utilisez la méthode GET /action/configfiles.

**Remarque** Avec GET /action/downloadconfigfile/{objId}, vous spécifiez généralement le nom du fichier comme identifiant de l'objet. Vous pouvez également spécifier l'identifiant de l'objet ConfigExportStatus associé au fichier.

### Procédure

**Étape 1** Obtenez une liste des fichiers de configuration sur le disque.

La liste des fichiers de configuration comprend les fichiers d'exportation et tous les fichiers que vous avez téléversés pour l'importation.

La commande curl ressemblerait à ce qui suit :

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

La réponse montrerait une liste d'items, chacun étant un fichier de configuration. Par exemple, la liste suivante affiche 2 fichiers. Notez que la valeur **id** (identifiant) affiche « default » (par défaut) pour tous les fichiers. Ignorez la valeur id (identifiant) et utilisez plutôt **diskFileName** (nomDeFichierDeDisque).

```
{
  "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/dernière version/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/dernière version/action/configfiles/default"
      }
    }
  ],
```


**Étape 2** Téléchargez le fichier utilisant le diskFileName comme l'identifiant de l'objet.

La commande curl ressemblerait à ce qui suit :

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

Le fichier est téléchargé dans votre dossier de téléchargement par défaut. Si vous utilisez la méthode GET à partir de l'explorateur API et que votre navigateur est configuré pour demander l'emplacement du téléchargement, vous serez invité à enregistrer le fichier.

Un téléchargement réussi résultera en un code de retour 200 et aucun corps de réponse.

# Modifier le fichier de configuration exporté

Après avoir téléchargé le fichier de configuration, vous pouvez le décompresser et ouvrir le fichier texte qui contient les objets. WordPad formatera le contenu de manière plus facile à lire que NotePad. Vous pouvez également utiliser d'autres éditeurs de texte installés sur votre dispositif, le cas échéant. Vous pouvez même créer votre propre fichier de configuration à partir de zéro, mais vous devrez quand même exporter la configuration pour comprendre la structure du fichier.

Les rubriques suivantes expliquent les exigences pour le fichier texte.

## Exigences minimales du fichier de configuration

Un fichier de configuration doit comporter les éléments minimaux suivants :

- Placez les objets dans le fichier entre [crochets]. Le fichier entier utilise la notation JSON standard et est un tableau d'objets.
- Enveloppez chaque objet dans des {accolades}.
- Utilisez des virgules pour séparer les objets dans le fichier de configuration. C'est-à-dire que l'accolade marquant la fin d'un objet doit être suivie d'une virgule, sauf pour l'objet final.
- Le premier objet du fichier doit être un objet de métadonnées. Le moyen le plus simple d'obtenir les bons attributs d'objet consiste à exporter la configuration à partir d'un dispositif du modèle souhaité. Par exemple, voici l'objet de métadonnées d'un dispositif Cisco Secure Firewall Threat Defense Virtual.

Avant d'importer le dispositif, vous pouvez modifier les types de configuration et d'exportation, et si vous le souhaitez, supprimer l'attribut generatedOn (généréLe).

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

- L'objet de métadonnées doit spécifier la valeur de type de configuration (configType) appropriée.
  - FULL_CONFIG : ce fichier texte comprend la configuration complète du dispositif.
  - DELTA_CONFIG : ce fichier texte comprend une configuration partielle, peut-être même quelques objets seulement.
- L'option exportType (typeD'Exportation) est l'une des options suivantes : FULL_EXPORT, PARTIAL_EXPORT, PENDING_CHANGE_EXPORT.
- Si vous effectuez une importation de configuration complète, l'objet de métadonnées doit spécifier les attributs suivants : hardwareModel (modèleDuMatériel), softwareVersion (versionDuLogiciel), apiVersion (versionDeL'Api).
- Vous pouvez écrire des objets sur une ou plusieurs lignes, mais ne placez pas de lignes vides ou de lignes de commentaires entre les attributs d'un objet. Les commentaires ne sont pas autorisés dans le fichier.
- Bien que les objets soient exportés en ordre de dépendance, où un objet référencé par un autre objet est défini en premier, il n'est pas obligatoire de maintenir cet ordre dans le fichier de configuration d'importation. Le système résoudra automatiquement les relations lors de l'importation, en supposant que les noms d'objet et les identifiants se résolvent correctement entre les objets dépendants.

## Structure de base des objets de classe d'enveloppe d'identité

Le fichier de configuration utilise des objets d'enveloppe d'identité pour définir tout objet ConfigEntity ou ManagementEntity qui peut être exporté ou importé. Voici la structure de base d'un objet d'enveloppe d'identité :

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

L'objet contient les attributs suivants :

- **type** (type) : la valeur est toujours **identitywrapper** (enveloppeD'Identité).
- **data** (données) : il s'agit de l'ensemble des paires attribut-valeur qui définissent l'objet à partir de la configuration, comme un objet réseau, une règle de contrôle d'accès, etc. Les attributs nécessaires à cette collecte dépendent du modèle du type d'objet et d'action spécifiques que vous effectuez. Mettez les paires attribut-valeur entre accolades. Séparez les attributs du tableau de données par des virgules.

- **parentName** (nomDuParent) : (si nécessaire.) Un nombre limité d'objets sont des ContainedObjects (ObjetsContenus), qui ont une relation avec un objet qui les contient. Cela inclut par exemple les critères d'accès, les règles NAT manuelles et les sous-interfaces. Pour ces éléments, le parentName (nomDuParent) spécifie le nom de l'objet le contenant (le parent). Précisez cet attribut pour les objets contenus. Ne le spécifiez pas pour les objets non contenus. Vous devrez peut-être également spécifier un index pour ces objets.

  Vous pouvez en fait omettre cet attribut si le parent est un objet unique (c'est-à-dire que vous ne pouvez pas en créer plusieurs), comme l'objet AccessPolicy (PolitiqueD'Accès), et le système sera en mesure de comprendre la référence.

- **oldName** (ancienNom) – (si nécessaire.) Si vous renommez un objet existant, vous pouvez spécifier l'ancien nom sur cet attribut et le nouveau nom dans l'attribut **name** des attributs de données. L'action doit être EDIT (MODIFIER) pour pouvoir utiliser cet attribut.

- **action** (action) : l'action à entreprendre par rapport à l'objet défini. Dans les exportations complètes, l'action est toujours **CREATE** (CRÉER). Pour les modifications en attente ou les exportations partielles, d'autres actions peuvent être **EDIT** (MODIFIER) ou **DELETE** (SUPPRIMER).

  Lorsque vous modifiez le fichier pour l'importation, précisez l'action souhaitée. Notez que si vous spécifiez CREATE (CRÉER), mais que l'objet existe déjà, l'action est remplacée par EDIT (MODIFIER); si l'objet n'existe pas, EDIT (MODIFIER) est remplacé par CREATE (CRÉER). L'action DELETE (SUPPRIMER) n'est pas modifiée. Les références aux objets sont résolues en fonction du type et du nom de l'objet, ou du type d'objet et de l'ancien nom, ou du type d'objet et du nom parent.

  - CREATE (CRÉER) : vous créez un nouvel objet. Vous devez spécifier les attributs de données requis lors de la publication d'un objet. Notez que si le **name** (nom) correspond à un objet existant du type spécifié, l'action passe automatiquement à EDIT (MODIFIER).

    Notez que si vous créez un objet et le référencez à partir d'autres objets, par exemple en définissant un objet réseau et en l'utilisant dans une règle d'accès, le **name** (nom) d'objet doit être le bon dans la référence.

  - EDIT (MODIFIER) : vous mettez à jour un objet. Vous devez spécifier les attributs de données requis lors de la mise en place d'un objet, à l'exception de la version et de l'identifiant. Le nom et le type d'objet sont utilisés pour déterminer l'objet à mettre à jour, et l'attribut de version est toujours ignoré.

  - DELETE (SUPPRIMER) : vous supprimez l'objet. Vous devez préciser les attributs **type** (type) et **name** (nom) dans les données de l'objet.

- **index** (index) : (facultatif; nombre entier.) Pour les objets qui font partie d'une liste ordonnée, comme les règles de contrôle d'accès et les règles NAT manuelles, il s'agit de la position de l'objet dans la politique. Si vous créez une nouvelle règle et que vous ne spécifiez pas de valeur d'index, la règle est ajoutée à la fin de la politique en tant que dernière règle. Si vous modifiez la règle, le système conservera la position existante de la règle.

## Exemple : Modifier un objet de réseau pour l'importation dans un dispositif différent

Chaque objet est structuré comme l'objet qui suit, un objet d'hôte de réseau qui définit l'adresse IP du serveur de journalisation du système :

```
{"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"}}
```

Supposons que vous ayez exporté cet objet à partir d'un dispositif et que vous souhaitez importer l'objet dans un autre dispositif, mais le nouveau dispositif doit utiliser un serveur de journalisation du système à une adresse différente, 192.168.5.15. Puisque vous allez créer un nouvel objet, supprimez les attributs **version** (version) et **id** (identifiant) de l'attribut data (données). Vous pouvez également supprimer **isSystemDefined** (estDéfiniParLeSystème) (dont la valeur par défaut est false [faux]) et **dnsResolution** (résolutionDuDns) (qui ne concerne qu'un objet FQDN uniquement). Le nouvel objet résultant ressemblerait à ce qui suit :

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

En haut du fichier, vous devez garder (ou ajouter) l'objet metadata (métadonnées). Vous pouvez également ajouter des retours à la ligne pour faciliter la lecture et la vérification du contenu du fichier. Ainsi, le fichier de configuration complet devrait ressembler à ce qui suit :

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

# Téléverser le fichier d'importation

Avant de pouvoir importer un fichier de configuration dans un dispositif, vous devez d'abord téléverser le fichier sur ce dernier. Vous pouvez téléverser des fichiers zip ou texte. Vous pouvez inclure des paquets et des profils client AnyConnect si vous utilisez un fichier zip.

Utilisez la ressource POST /action/uploadconfigfile pour téléverser le fichier. Le nom a une longueur maximale de 60 caractères.

- Si vous utilisez cette méthode à partir de l'explorateur d'interface de protocole d'application, cliquez sur le bouton **Choose File** (Choisir un fichier) à côté de l'attribut **fileToUpload** (fichierÀTéléverser) pour sélectionner le fichier dans le lecteur de votre ordinateur.

- Si vous utilisez la méthode de votre propre programme, la charge utile de la demande doit contenir un seul file-item (élément-de-fichier) avec un champ file-name (nom-de-fichier). L'extension de file-name (nom-de-fichier) doit être .txt ou .zip et le format du contenu du fichier réel doit être cohérent avec l'extension de fichier.

La commande curl ressemblerait à ce qui suit :

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

Un transfert réussi génère un code de retour 200 et un corps de réponse semblable à ce qui suit, qui affiche le nom du fichier sur le système Défense contre les menaces (**diskFileName**), dont vous avez besoin pour la tâche d'importation.

```
{
  "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/dernière version/action/uploadconfigfile/default"
  }
}
```

# Importer la configuration et vérifier l'état de la tâche

Après avoir téléversé un fichier de configuration dans le système Défense contre les menaces, vous pouvez importer les objets définis dans le fichier de configuration dans la configuration de Défense contre les menaces. Utilisez la méthode POST /action/configimport.

Lorsque vous importez des objets, vous avez également la possibilité de définir les objets directement dans la commande d'importation plutôt que dans un fichier de configuration. Cependant, vous ne devriez définir directement des objets que dans les cas où vous importez un petit nombre de modifications, comme à un ou deux objets réseau seulement.

### Procédure

**Étape 1** Créez le corps de l'objet JSON pour la tâche d'importation.

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

```
{
  "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"
}
```

Les attributs sont :

- **diskFileName** (nomDuFichierDeDisque) : le nom du fichier de configuration zip ou .txt à importer.
- **encryptedKey** (cléChiffrée) : la clé utilisée pour chiffrer le fichier zip, le cas échéant. Ne précisez pas de clé si le fichier de configuration n'est pas chiffré.
- **preserveConfigFile** : (facultatif.) Indique s'il faut conserver la copie du fichier de configuration importé sur le disque de Défense contre les menaces après une tâche d'importation réussie. Précisez true (vrai) pour conserver le fichier, false (faux) pour que le fichier soit supprimé du disque de Défense contre les menaces. La valeur par défaut est false (faux).
- **autoDeploy** (déploiementAutomatique) : (facultatif.) Indique s'il faut démarrer automatiquement une tâche de déploiement si l'importation est réussie. Les objets importés sont mis en attente avant que les modifications entrent en vigueur, et ces dernières ne seront actives que lorsque vous les déploierez. Précisez « true » (vrai) pour démarrer la tâche de déploiement automatiquement. Si vous indiquez false (faux), vous devrez déployer manuellement vos modifications. La valeur par défaut est false (faux).
- **allowPendingChange** (autoriserLesModificationsEnAttente) : (facultatif.) Indique s'il faut autoriser la tâche d'importation à démarrer si des modifications existantes sont en attente. Si vous définissez cet attribut sur « true » (vrai) et autoDeploy (déploiementAutomatique) sur « true » (vrai), la tâche de déploiement automatique inclura toutes les modifications, préexistantes et importées. Si vous définissez cet attribut sur false (faux), la tâche d'importation ne s'exécutera pas si des modifications sont en attente. La valeur par défaut est false (faux).
- **excludeEntities** (exclureEntités) : (facultatif.) Une liste de chaînes correspondant à des objets qui identifient les objets qui ne doivent pas être importés. Vous devez préciser cet attribut uniquement si le fichier d'importation comprend des éléments que vous ne souhaitez pas importer (c'est-à-dire que vous avez décidé de ne pas les supprimer du fichier que vous avez téléversé). Chaque élément de cette liste a un schéma ressemblant à « **id=***uuid-value* », « **type**=*object-type* » ou « **name**=*object-name* ». Les objets d'entrée correspondant à l'un de ces schémas seront exclus de l'importation.

  Le **type** peut être soit une entité leaf, telle qu'un networkobject (objetréseau), ou un alias d'un ensemble de types leaf. Voici quelques alias de types typiques : network (NetworkObject et NetworkObjectGroup), port (tous les types de port, de protocole et de groupe TCP/UDP/ICMP), url (objets et groupes URL), ikepolicy (politiques IKE V1/V2), ikeproposal (propositions Ike V1/V2), identitysource (toutes les sources d'identité), certificate (tous les types de certificats), object (tous les types d'objets/groupes qui seraient répertoriés dans le gestionnaire d'appareil sur la page Objects), interface (toutes les interfaces réseau, s2svpn (tous les types de VPN site à site), ravpn (tous les types de VPN RA), vpn (s2svpn et ravpn).

Par exemple, pour exclure de l'importation tous les objets réseau et deux autres objets identifiés par le nom myobj et un UUID, spécifiez :

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

- **inputEntities** (entitésDeSaisie) : si vous avez un petit nombre d'objets à importer, vous pouvez les définir dans la liste d'objets d'entrée d'entrée plutôt que dans un fichier de configuration. Pour utiliser cet attribut, vous ne pouvez pas inclure l'attribut diskFileName, sinon vous devrez le définir sur « null » (nul).
- **jobName** (nomDeLaTâche) : (facultatif.) Un nom pour la tâche d'exportation. Donner un nom à la tâche d'exportation permet de la retrouver plus facilement lorsque vous récupérez le statut de la tâche.
- **type** (type) : le type de tâche, qui est toujours **scheduleconfigimport** (importationdelaconfigurationducalendrier).

**Exemple :**

L'exemple suivant importe le fichier de configuration nommé import-1.txt :

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

**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 '{ \
   "diskFileName": "import-2.txt", \
   "preserveConfigFile": true, \
   "autoDeploy": true, \
   "allowPendingChange": true, \
   "type": "scheduleconfigimport" \
 }' 'https://10.89.5.38/api/fdm/dernière version/action/configimport'
```

**Étape 3** Vérifiez la réponse.

Vous devriez obtenir un code réponse de 200. Si vous publiez l'objet JSON minimum, le corps de la réponse ressemblera à ce qui suit : Si vous spécifiez une clé de chiffrement, elle sera masquée dans la réponse.

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

**Étape 4** Utilisez GET /jobs/configimportstatus pour vérifier l'état de la tâche d'importation.

Vous pouvez également utiliser GET /jobs/configimportstatus/{objId} pour obtenir l'état d'une tâche d'importation. Pour objId (idObj), utilisez la valeur jobHistoryUuid du corps de la réponse à votre appel POST /action/configimport.

La commande curl ressemblerait à ce qui suit :

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

Le corps de la réponse pourrait ressembler à ce qui suit pour une importation réussie. Si l'importation échoue, vous devrez peut-être modifier le fichier pour corriger les erreurs de formatage ou de contenu, puis réessayer.

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

### Prochaine étape

Si vous définissez la valeur autoDeploy (déploiementAutomatique) sur false (faux), vous devrez exécuter une tâche de déploiement pour intégrer les modifications importées. Utilisez la méthode POST /operational/deploy. Si vous avez défini la valeur sur true (vrai), la configuration devrait avoir été déployée avec succès. Dans le gestionnaire d'appareil ou l'API (GET /operational/auditevents), vous pouvez vérifier le journal d'audit. La tâche de déploiement s'intitule « Post Configuration Import Deployment » (Déploiement après l'importation de la configuration).

**Remarque** Certaines fonctionnalités nécessitent des licences particulières. Par exemple, un dispositif doit avoir une licence pour toutes les fonctionnalités VPN d'accès à distance. Toutefois, le processus d'importation ne valide pas les licences. Par conséquent, si vous importez des objets pour une fonctionnalité contrôlée par licence sur un dispositif qui ne dispose pas de la licence requise, la tâche de déploiement échouera. Si vous rencontrez ce problème, attribuez les licences requises au dispositif ou supprimez les objets.

# Supprimer les fichiers d'importation/exportation non requis

Si vous n'avez plus besoin d'un fichier de configuration, qu'il soit créé par une tâche d'exportation ou que vous l'ayez téléversé pour l'importation de la configuration, vous pouvez le supprimer.

Utilisez la méthode DELETE /action/configfiles/{objId}, en utilisant le nom du fichier comme valeur objId.

Par exemple, pour supprimer le fichier nommé export-config-2.zip, la commande curl serait la suivante :

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

La commande a réussi si vous recevez un code de retour 204 et aucun corps de réponse.

Vous pouvez utiliser GET /action/configfiles pour confirmer que le fichier a été supprimé.

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.