À propos de l'API REST Cisco Secure Firewall Threat Defense

Vous pouvez utiliser l'interface de programmation d'applications (API) de transfert d'état de représentation (REST) Cisco Secure Firewall Threat Defense, par l'entremise du protocole HTTPS, pour interagir avec le dispositif Défense contre les menaces par le biais d'un programme client. L'API REST utilise le format JSON (JavaScript Object Notation) pour représenter les objets.

Cisco Secure Firewall device manager comprend un explorateur d'interface de protocole d'application qui explique toutes les ressources et les objets JSON disponibles à être utilisés en programmation. L'explorateur fournit des informations détaillées sur les paires attribut-valeur dans chaque objet, et vous pouvez tester les différentes méthodes HTTP pour vous assurer de bien comprendre le codage requis pour utiliser chaque ressource. L'explorateur d'interface de protocole d'application fournit également des exemples des URL requises pour chaque ressource.

Vous pouvez également trouver des informations de référence et des exemples en ligne sur le site https://developer.cisco.com/site/ftd-api-reference/.

L'API a son propre numéro de version. Il n'est pas garanti qu'un client conçu pour une version de l'API fonctionnera pour une version future sans erreur ou sans nécessiter de modifications à votre programme.

Public cible de ce guide de programmation

Ce guide a été écrit avec la présomption que vous avez une connaissance générale de la programmation et une compréhension précise des API REST et de JSON. Si vous débutez dans l'utilisation de ces technologies, veuillez d'abord lire un guide général sur les API REST.

Méthodes HTTP prises en charge

Vous ne pouvez utiliser que les méthodes HTTP suivantes : les autres méthodes ne sont pas prises en charge.

- GET : pour lire les données du système.
- POST : pour créer des objets.
- PUT : pour modifier les objets existants. Lorsque vous utilisez PUT, vous devez inclure l'objet JSON entier. Vous ne pouvez pas mettre à jour de manière sélective les attributs individuels d'un objet.
- DELETE : pour supprimer un objet défini par l'utilisateur.

L'URL de base pour l'API

La façon la plus simple de déterminer l'URL de base pour un dispositif Défense contre les menaces donné est d'essayer une méthode GET dans l'explorateur d'interface de protocole d'application et de supprimer simplement la partie objet de l'URL du résultat.

Par exemple, vous pouvez appeler une méthode GET /object/networks et voir quelque chose de similaire à ce qui suit dans la sortie renvoyée sous Request URL (URL de demande) :

```
https://ftd.example.com/api/fdm/v1/object/networks
```

La partie du nom du serveur de l'URL est le nom d'hôte ou l'adresse IP du dispositif Défense contre les menaces et sera différente pour votre dispositif à la place de « ftd.example.com ». Dans cet exemple, vous devez supprimer /object/networks du chemin pour obtenir l'URL de base :

```
https://ftd.example.com/api/fdm/v1/
```

Tous les appels de ressource utilisent cette URL comme base pour l'URL de demande.

Si vous avez modifié le port de données HTTPS, vous devez inclure le port personnalisé dans l'URL. Par exemple, si vous avez changé le port en 4443 : https://ftd.example.com:4443/api/fdm/v1/

Le « v » dans l'URL représente la version de l'API, et cela change généralement avec la version du logiciel. Par exemple, la version de l'API pour Défense contre les menaces, version 6.3.0, est v2, donc l'URL de base serait :

```
https://ftd.example.com/api/fdm/v2/
```

Remarque À partir de la version 6.4 de Défense contre les menaces, vous pouvez éviter d'avoir à mettre à jour le chemin dans vos appels d'API en utilisant **latest** (dernier) au lieu de l'élément v dans le chemin. Par exemple, https://ftd.example.com/api/fdm/latest/. L'alias **latest** (dernier) représente la dernière version d'API prise en charge par le dispositif.

Dans l'explorateur d'interface de protocole d'application, si vous faites défiler la page vers le bas de la page, vous pouvez voir des informations sur l'URL de base (sans le nom du serveur) et la version de l'API.

Sécurisation des communications SSL/TLS pour l'API REST

Les appareils Défense contre les menaces sont livrés avec un certificat autosigné pour que vous puissiez l'utiliser pour initier des communications HTTPS. Cependant, comme le certificat n'est pas signé par une autorité de certification (AC) connue, toute tentative d'accès par SSL/TLS fera en sorte que la connexion sera considérée comme non sécurisée.

Au moment de la connexion à un navigateur, vous serez invité à accepter le certificat autosigné, mais une commande comme la commande « curl » rejettera le certificat. Dans le cas de la commande « curl », vous pouvez contourner l'échec de la vérification du certificat en ajoutant le mot-clé **--insecure**. Par exemple :

```
curl --insecure -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/versions'
```

L'une des premières choses que vous devez faire est d'obtenir un certificat de dispositif signé par une autorité de certification pour le dispositif Défense contre les menaces. Ensuite, à l'aide du gestionnaire d'appareil ou de l'API, attribuez ce certificat comme certificat de gestion. La vérification du certificat SSL/TLS ne devrait alors pas échouer et vous n'aurez pas à utiliser des communications non sécurisées dans vos appels d'API.

Procédure

Étape 1 Téléversez le certificat de dispositif signé par une autorité de certification à l'aide de la ressource **POST /objet/internalcertificates**.

Étape 2 Faites de ce certificat le certificat de gestion à l'aide de la ressource **PUT /devicesettings/default/webuicertificates/{objId}**.

Utilisez la ressource **GET /devicesettings/default/webuicertificates** pour déterminer l'ID d'objet du certificat d'interface utilisateur Web.

Étape 3 Déployez les modifications à l'aide de la ressource **POST /operational/deploy**.

Déterminer les versions de l'API prises en charge

Vous pouvez déterminer quelles versions de l'API sont prises en charge sur un dispositif à l'aide de la méthode GET /api/versions (ApiVersions). Cette méthode ne nécessite pas d'authentification et n'inclut pas non plus d'élément indiquant la version dans le chemin. Par exemple :

```
curl -X GET --header 'Accept: application/json' 'https://ftd.example.com/api/versions'
```

Remplacez « ftd.example.com » par le nom d'hôte ou l'adresse IP du dispositif défense contre les menaces.

Cette méthode renvoie une liste de versions d'API que vous pouvez utiliser. Par exemple :

```
{
    "supportedVersions":["v3", "latest"]
}
```

Les chaînes de version sont les mêmes que celles que vous utilisez dans l'URL pour les appels d'API ultérieurs. Si vous utilisez **latest** (dernier) au lieu de l'identifiant de version spécifique, vous pouvez éviter d'avoir à mettre à jour vos appels pour les versions ultérieures. Cependant, utiliser cette technique ne permet pas de tenir compte des modifications des modèles d'objet utilisés dans vos appels, qui pourraient devoir être adaptés d'une version à l'autre.

En règle générale, votre prochaine étape consistera à obtenir un jeton d'accès, comme décrit dans Authentifier votre client API REST à l'aide d'OAuth.

Rétrocompatibilité des versions de l'API

La version de l'API défense contre les menaceschange avec chaque version majeure du logiciel défense contre les menaces. Les nouvelles fonctionnalités ont une incidence sur les appels d'API pour les fonctionnalités en cours d'ajout ou de modification.

Cependant, de nombreuses fonctionnalités ne changent pas d'une version à l'autre. Par exemple, les API liées aux objets réseau et port restent souvent inchangées dans une nouvelle version.

À partir de la version 6.7 de défense contre les menaces, si un modèle de ressource d'API pour une fonctionnalité ne change pas entre les versions, l'API défense contre les menacespeut accepter les appels basés sur l'ancienne version de l'API. Même si le modèle de fonctionnalité a changé, s'il existe un moyen logique de convertir l'ancien modèle au nouveau modèle, l'ancien modèle peut fonctionner. Par exemple, un appel v5 peut être accepté sur un système v6. Si vous utilisez « latest » (dernier) comme numéro de version dans vos appels, ces appels « anciens » sont interprétés comme un appel v6 dans ce scénario. Par conséquent, votre utilisation de la compatibilité ascendante dépend de la façon dont vous structurez vos appels d'API.

Si un modèle de fonctionnalité a changé entre les versions d'API d'une manière qui empêche la rétrocompatibilité, vous recevrez un message d'erreur et vous devrez vérifier ces erreurs et mettre à jour votre code pour ces appels en particulier.

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.