gRPC トンネル

gRPC トンネルについて

Cisco NX-OSは、「gPRC トンネル」をサポートします。これは、アプリケーションレイヤトンネルの特定の実装で、ファイアウォールなどで分離されたネットワーク間での外部通信を可能にします。

一般的なネットワーク展開では、ファイアウォールを使用して、ネットワークをファイアウォールの「外部」と「内部」に大まかに分離します。そうした場合、たとえばgNMI では、「gnmi クライアント（コントローラ）」は通常ファイアウォールの外部にあり、「gnmi サーバ（ネットワーキングスイッチ）」は内部にあります。したがって、gNMI 接続はスイッチへのインバウンド RPC と呼ばれ、通常は「ダイヤルイン」と呼ばれます。

このような「ダイヤルイン」には、次の 2 つの制限があります。

- **ファイアウォールのプロビジョニング**：

  「ダイヤルイン」では、gNMI 接続が通過できるように、特定のホスト、アドレス、ポートなどに関連したファイアウォールルールを指定して、ファイアウォールに穴をあける必要があります。

- **前提条件のネットワークインベントリ**：

  特定の接続先に「ダイヤルイン」するには、最初に勧誘手順が必要です。ユーザーは手動またはプログラムで、gNMI サーバーのホスト情報を収集します。

  そうして初めて、クライアントはそれらのサーバーに「ダイヤルイン」するためのアドレス/ポートを知ることができます。

grpc-tunnel は、これら 2 つの制限を回避することを目的として、逆の「ダイヤルアウト」アプローチによりトンネルを確立します。grpc トンネルの詳細については、 gRPC Tunnel の外部ドキュメントを参照してください。

注意事項と制約事項

gRPC トンネルには、次の注意事項と制約事項があります。

- トンネルのターゲット識別子を割り当てるときの命名規則は、完全にユーザーに任されています。
- ユーザーは、ターゲット識別子の命名規則が一意であることを確認する必要があります。自動展開ワークフローでターゲット識別子の一意性を扱うようにすることをお勧めします。
- Cisco NX-OS は、最大 8 つのトンネル構成をサポートします。

トピック 2.1

gRPC トンネルの設定

認証なしの gRPC トンネルの構成

この手順では、サーバまたはクライアント認証なしに、gRPC トンネルを有効にして構成する方法について説明します。これは主に実験用です。

始める前に

gRPC トンネルは、さまざまなネットワークトラフィックを転送できることが想定されている、透明性のないトンネルです。ただし、Cisco NX-OS における主なユースケースは、gNMI/gNOI 要求のプロキシを行うことです。その場合は、grpcエージェントを適切に設定する必要があります。gRPC エージェントのプログラミングガイドを参照してください。

手順の概要

1. **configure terminal**
2. **[no] feature grpctunnel**
3. **[no] grpctunnel destination**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch-1# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**[no] feature grpctunnel** 例： `switch-1# feature grpctunnel`	grpc-tunnel 機能を有効または無効にします。
ステップ 3	**[no] grpctunnel destination** 例： `switch-1# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI use-vrf management`	トンネル接続を構成します。トンネル接続を削除するには、このコマンドの no 形式を使用します。 • **destination**：（タイプ：IPv4/IPv6 アドレスまたはホスト名文字列）トンネル サーバーの IP アドレスまたはホスト名。ホスト名が指定されている場合は、有効なネームサーバー構成が必要です。 • **port**：（タイプ：tcp port）トンネル サーバーのポート番号。 • **target**：（タイプ：文字列、最大 64 バイト）ターゲット ID は文字列です。ユーザーが ID を予約済みキーワード「HOSTNAME」に設定すると、スイッチはスイッチのホスト名でターゲットを置き換えます。 • **type**：（タイプ：文字列、制限 64 バイト）タイプは、10.3.2F リリースの GNMI_GNOI のみをサポートします。 • **use-vrf**：（タイプ：文字列）スイッチが grpc トンネル セッションのダイヤル アウトに使用する VRF 名文字列。 • **（オプション）source-interface**：（タイプ：インターフェイス名文字列）source-interface は、トンネル確立の際の出力送信元 IP アドレスを決定するために使用されます。設定すると、スイッチは、インターフェイスの最初の ipv6 グローバルユニキャストアドレスを選択します。それ以外の場合は、インターフェイスの ipv4 ユニキャスト アドレスを選択します。この設定

	コマンドまたはアクション	目的
		は、ループバックおよび svi インターフェイスのみをサポートします。インターフェイスは、Lo10、Vlan100 などの短縮名形式で指定する必要があります。 • （オプション）**target-vrf**：（タイプ：文字列）ローカル grpc サーバー ターゲットに到達するため、指定した VRF 名を使用します。指定しない場合は、vrf パラメータと同じ名前を使用します。たとえば、grpctunnel を use-vrf foo のように指定したとすると、target-vrf bar は、スイッチが vrf foo 内の外部トンネルサーバへの接続を確立するものの、着信 grpc 要求は vrf bar 内に存在するローカルスイッチ grpc サーバに転送することを意味します。

トラストポイントを使用した gRPC トンネルの構成

gRPC トンネルの場合、 Cisco NX-OS デバイスが指定された外部 gRPC トンネル接続先への接続を開始することに注意してください。ユーザーは、このようなアウトバウンド接続を保護するためにトラストポイント/証明書を構成できます。

- 「cert」オプションを使用したサーバ認証：これにより、外部接続先の証明書がスイッチに構成されます。構成された証明書がリモートトンネルと一致しない場合、スイッチは接続を拒否します。
- 「client-cert」オプションを使用したクライアント認証：これにより、スイッチのアイデンティティ証明書が構成されます。スイッチが一致する証明書を提示できない場合、リモートトンネルサーバはスイッチからの接続を拒否します。
- 相互認証は、「cert」認証と「client-cert」認証の両方を組み合わせたものです。

以下の手順では、手順 1 ～ 3 は「サーバ認証」をインポートすることを意図し、手順 4 ～ 5 は「クライアント認証」をインポートすることを意図しています。ユーザーは、いずれかまたは両方を有効にする適切な組み合わせを決定できます。

（注） クライアント認証用のルート証明書を構成または削除すると、gRPC トンネルは、リモート接続先への接続を再起動します。

（注） クライアントの証明書が中間 CA によって署名されているが、上記の構成からインポートされたルート CA によって直接署名されていない場合、grpc トンネル証明書（ステップ 5）は、ユーザー、中間 CA 証明書、およびルート CA 証明書を含む完全な証明書チェーンを提供する必要があります。

始める前に

サーバ認証に必要な証明書ファイルを準備して署名します。これは gRPC トンネルに固有のものではないため、既存のトラストポイントファイルを再利用できます。

このセクションは、特に「cert」および「client-cert」オプションの使用方法を明確にすることを意図しています。前のセクションで説明したオプションは、これら 2 つのオプションと自由に組み合わせることができます。

手順の概要

1. **configure terminal**
2. （任意） **crypto ca trustpoint** <*tunnel-trustpoint*>
3. （任意） **rsakeypair** <*client-key*>
4. （任意） **crypto ca authenticate** <*tunnel-trustpoint*>
5. （任意） **crypto ca trustpoint** <tunnel-client-trustpoint>
6. （任意） **crypto ca import** <*tunnel-client-trustpoint*> **pkcs12 bootflash** *:*<*ca-file*> <*pkcs-password*>
7. **[no] grpctunnel destination …**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	（任意） **crypto ca trustpoint** <*tunnel-trustpoint*> 例： `switch# crypto ca trustpoint tunnel_trustpoint`	トンネルサーバ認証用のトラストポイントを作成します。
ステップ 3	（任意） **rsakeypair** <*client-key*> 例： `switch# rsakeypar key`	クライアントトラストポイントの rsa キーペアを生成します。

	コマンドまたはアクション	目的
ステップ 4	（任意） **crypto ca authenticate** <*tunnel-trustpoint*> 例： `switch# crypto ca authenticate tunnel_trustpoint`	トンネルサーバ証明書をインポートします。この手順では、手動でコピーして貼り付ける必要があります。指示に従ってください。
ステップ 5	（任意） **crypto ca trustpoint** <tunnel-client-trustpoint> 例： `switch# crypto ca trustpoint tunnel_client_trustpoint`	トンネルクライアント認証用のトラストポイントを作成します。
ステップ 6	（任意） **crypto ca import** <*tunnel-client-trustpoint*> **pkcs12 bootflash** *:<ca-file> <pkcs-password>* 例： `switch# crypto ca import tunnel_client_trustpoint pkcs12 bootflash:ca.pfx test`	pkcs12 ファイルをトラストポイントにインポートします。
ステップ 7	**[no] grpctunnel destination …** 例： `switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI cert tunnel_trustpoint client-cert tunnel_client_trustpoint …`	トンネル接続を構成します。トンネル接続を削除するには、このコマンドの no 形式を使用します。 • **[オプション] cert** -（タイプ：文字列）トンネルサーバ証明書を保持するトラストポイント。指定しない場合、サーバーの検証はスキップされます。 • （**オプション**）**client-cert**：（タイプ：文字列）クライアント証明書を保持するトラストポイント。指定した場合、 スイッチはトンネル サーバーとの相互認証を実行します。

VRF による gRPC トンネルの構成

以下の手順では、手順 1 ～ 3 は「サーバ認証」をインポートすることを意図し、手順 4 ～ 5 は「クライアント認証」をインポートすることを意図しています。ユーザーは、いずれかまたは両方を有効にする適切な組み合わせを決定できます。

始める前に

このセクションでは、特に「use-vrf」オプションと「target-vrf」オプションの使用方法を明確にします。前のセクションで説明したオプションは、これら 2 つのオプションと自由に組み合わせることができます。

手順の概要

1. **configure terminal**
2. **[no] feature grpctunnel**

3. **[no] grpctunnel destination …**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch-1# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**[no] feature grpctunnel** 例： `switch(config)# feature grpctunnel`	grpc-tunnel 機能を有効または無効にします。
ステップ 3	**[no] grpctunnel destination …** 例： `switch(config)# switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI use-vrf management target-vrf default`	トンネル接続を構成します。トンネル接続を削除するには、このコマンドの no 形式を使用します。 • **use-vrf**：（タイプ：文字列）スイッチが grpc トンネル セッションのダイヤル アウトに使用する VRF 名文字列。 • **（オプション）target-vrf**：（タイプ：文字列）ローカル grpc サーバー ターゲットに到達するため、指定した VRF 名を使用します。指定しない場合は、vrf パラメータと同じ名前を使用します。たとえば、grpctunnel を use-vrf foo のように指定したとすると、target-vrf bar は、スイッチが vrf foo 内の外部トンネルサーバへの接続を確立するものの、着信 grpc 要求は vrf bar 内に存在するローカルスイッチ grpc サーバに転送することを意味します。

例

このセクションでは、トンネルの使用方法を示すいくつかの構成例を示します。

認証なし

次の手順では、サーバ検証を行わずにトンネルの接続先を構成する方法について説明します。

```
switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI
use-vrf management
switch(config)# grpctunnel destination server.foo.com port 8000 target test2 type GNMI_GNOI
 use-vrf management
```

この例では、ユーザーは2つのトンネル接続先「1.1.1.1:8000」と「server.foo.com:8000」を、それぞれターゲット「test1」と「test2」として構成します。接続は、管理名前空間を介して開始されます。

サーバ認証あり

次の手順では、サーバ検証を行ってトンネルの接続先を構成する方法について説明します。

次のコマンドを実行して、トラストポイントにサーバ証明書をインポートします。

```
switch(config)# crypto ca trustpoint tunnel_server_trustpoint switch(config-trustpoint)#
 crypto ca authenticate tunnel_server_trustpoint
input (cut & paste) CA certificate (chain) in PEM format; end the input with a line
containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC3TCCAcWgAwIBAgIJAO4xEeL+IrpuMA0GCSqGSIb3DQEBCwUAMBcxFTATBgNV
BAMMDHNqYy1hZHMtNjAxNDAeFw0yMjA1MjYwMDE4MzBaFw0zMjA1MjMwMDE4MzBa
MBcxFTATBgNVBAMMDHNqYy1hZHMtNjAxNDCCASIwDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBALudrG824XmW/4+BNd632CT3x47akV0QfjwAU1xBDScpAw9brERO
YTLP9BxInbA+WAS+zGql6nmBoZxbqZZL/NVD8ltLKYJJxtDQHJkqdX2lURnMUFr2
9pyJQtuh/udq9hp8zGcEpbPayfIdHCnZqraWMLvk1W0mqAa7ek0iizIZNwKmU3oR
7CGQOxi8aMsAfH5iBsRTNURFdaXdJYTOjry0il+jBKT21F2Z3vGcB7ddTt+I7qrd
GjJs4BI4a22Y3usYb/dnsEa0ZCFTFIq6Y2Pwc3DOuKalUhujSqisqfMDuqC34ATw
kWwLnHDWVu0iVaWndy3uvQZKDNv/bIIuoo8CAwEAAaMsMCowFwYDVR0RBBAwDoIM
c2pjLWFkcy02MDE0MA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEB
AIjNgq/paYfPtHDe9PlZKzrmGz+UlUAx8saj2WHtrKgBj48J6fYvz1yTPWLKMPct
/5y+nhia6gRlV/navFcpiUUpQGpoZQnaa40/nkBMDvVxnTu6l9UC0WUAyTh2l7ec
BriY8yq3elpQWHZS4KRNmBH8fuviAv4f0fzOAuNGeIuv7UGnfA8Ed/q/Z3frQxOI
qNXr3vBBTptYTLwdrRM0axagL6waZgZyTFfFHpIXBPEtsXKb/5GuP4+nqXvtfkfe
d6P9jA4BKA/e6Gu6NAR0JMOdmJeEFjMbg+uu8jghcRTcwRsGeb9DqPUL+5IsVg3a dKMaZxyQFiRz0LyTqQtZmE0=
 -----END CERTIFICATE-----
END OF INPUT
Fingerprint(s): SHA1 Fingerprint=D4:9D:79:5B:8B:38:D6:50:6D:46:89:A8:C4:41:AB:
C9:D9:9F:D1:66
Do you accept this certificate? [yes/no]:yes
```

その後、次のコマンドを実行して、トンネルの接続先を構成します。

また、show コマンドを使用して設定をディスプレイします。

```
switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI
use-vrf
management cert tunnel_server_trustpoint

switch(config)# show system internal dme running-config all dn sys/grpctunnel {
"grpctunnelInst": {
"attributes": {
"childAction": "",
"dn": "sys/grpctunnel",
"modTs": "2022-12-02T12:57:37.891+00:00",
"status": ""
},
"children": [
{
"grpctunnelTunnelMgr": {
"attributes": {
"childAction": "",
"dn": "sys/grpctunnel/tunnelmgr",
"modTs": "2022-12-02T12:57:37.891+00:00",
"status": ""
},
"children": [
```

```
{
"grpctunnelTunnel": {
"attributes": {
"cert": "tunnel_server_trustpoint",
"certClient": "",
"childAction": "", "dest": "1.1.1.1", "dn":
"sys/grpctunnel/tunnelmgr/tunnel-[1.1.1.1]-port-[8000]-target-[test1]-type-[GNMI_GNOI]-vrf-[management]",
"modTs": "2022-12-05T10:09:45.163+00:00",
"port": "8000",
"srcIf": "unspecified",
"status": "",
"targetId": "test1",
"targetType": "GNMI_GNOI",
"targetVrf": "",
"vrf": "management"
}
}
}
]
}
}
]
}
}
```


クライアント認証あり

次の手順では、クライアント検証を行ってトンネルの接続先を構成する方法について説明します。

次の手順では、サーバ検証を行わずにトンネルの接続先を構成する方法について説明します。

```
switch(config)# crypto ca trustpoint tunnel_client_trustpoint
switch(config)# crypto ca import tunnel_client_trustpoint pkcs12 bootflash://ca.pfx test
```

その後、次のコマンドを実行して、トンネルの接続先を構成します。

また、show コマンドを使用して設定をディスプレイします。

```
switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI
use-vrf
management client-cert tunnel_client_trustpoint

switch(config)# show system internal dme running-config all dn sys/grpctunnel {
"grpctunnelInst": {
"attributes": {
"childAction": "",
"dn": "sys/grpctunnel",
"modTs": "2022-12-02T12:57:37.891+00:00",
"status": ""
},
"children": [
{
"grpctunnelTunnelMgr": {
"attributes": {
"childAction": "",
"dn": "sys/grpctunnel/tunnelmgr",
"modTs": "2022-12-02T12:57:37.891+00:00",
"status": ""
},
"children": [
{
```

```
"grpctunnelTunnel": {
"attributes": {
"cert": "",
"certClient": "tunnel_client_trustpoint ",
"childAction": "", "dest": "1.1.1.1", "dn":
"sys/grpctunnel/tunnelmgr/tunnel-[1.1.1.1]-port-[8000]-target-[test1]-type-[GNMI_GNOI]-vrf-[management]",
"modTs": "2022-12-05T10:09:45.163+00:00",
"port": "8000",
"srcIf": "unspecified",
"status": "",
"targetId": "test1",
"targetType": "GNMI_GNOI",
"targetVrf": "",
"vrf": "management"
}
}
}
]
}
}
]
}
}
```


VRF あり

「use-vrf」と「target-vrf」構成を組み合わせると、導入の柔軟性は高くなりますが、混乱が生じる可能性もあります。

次の違いに留意してください。

- use-vrf：リモートトンネルの接続先に到達する方法。
- target-vrf：スイッチの内部サービスにトンネル トラフィックを転送/リレーする方法。

次のシナリオの例を参照してください。

- リモートトンネルサーバは、「管理」vrfを経由して到達可能です。スイッチがトンネル内の gNMI 接続を受信すると、スイッチは gnmi「管理」サーバに転送します。

  ```
  grpctunnel destination server1 port 9000 target target2 type GNMI_GNOI vrf management
  ```

- ローカル grpcエージェントがデフォルトの vrf で実行されている間、リモートトンネルサーバは「管理」vrfを経由して到達可能です。次の構成では、スイッチがトンネル内の gNMI 接続を受信すると、スイッチは gnmi 要求をデフォルト vrf に送信します。

  ```
  grpc use-vrf default
  grpctunnel destination server1 port 9000 target target2 type GNMI_GNOI use-vrf
  management target-vrf default
  ```

- リモートトンネルサーバとローカル grpc エージェントの両方が、デフォルトの vrf で実行されます。

  ```
  grpc use-vrf default
  grpctunnel destination server1 port 9000 target target2 type GNMI_GNOI use-vrf default
  ```

- リモートトンネルサーバは「デフォルト」vrfを経由して到達可能ですが、ローカル grpc エージェントは「test」vrf で実行されています。次の構成では、スイッチがトンネル内の gNMI 接続を受信すると、スイッチは gnmi 要求を test vrf に送信します。

```
grpc use-vrf test
grpctunnel destination server1 port 9000 target target2 type GNMI_GNOI vrf default
local-vrf test
```

- この場合、ローカル grpc エージェントが「abc」vrfで実行されている間、リモートトンネルサーバは「デフォルト」vrfを経由して到達可能です。次の構成では、スイッチがトンネル内の gNMI 接続を受信すると、スイッチは gnmi 要求を test vrf に送信するため、接続は機能しません。これは前方参照として扱うことができます。grpc config を「grpc use-vrf test」に変更すると、接続は機能するようになります。

```
grpc use-vrf abc
grpctunnel destination server1 port 9000 target target2 type GNMI_GNOI vrf default
local-vrf test
```

障害対応

機能ステータスの確認

- Cisco NX-OS で、**show feature grpctunnel** コマンドを入力してエージェントの構成を確認します。
- gRPC トンネルのステータスを表示するには、**show feature** コマンドを使用します。

```
switch-1# show feature | grep grpctunnel
restconf 1 enabled
switch-1#
```

gRPC トンネルのデバッグ

トンネルのステータスを表示する一連の show コマンドがあります。

コマンドの表示

トンネルの構成/ステータスを確認するには、次のコマンドを入力します。

コマンド	説明

show grpctunnel internal sessions [all] { summary | detail } }	トンネルのステータスを表示しま • 「sessions」オプションは、ト を表示します。 「all」オプションを使用すると、終 終了したセッションについては、 を保持します。
debug grpctunnel events all	CLI EXEC モードで実行します。 これにより、コンソールにデバッ

出力例

show grpctunnel internal sessions summary

```
=============
gRPC Tunnel
=============
Restart Count : 1
* - history
Destination                                                Target/Type
 Cnt Retry  Cnt Sess   Status/Error
---------------------------------------------------------- ------------------------------
 ---------- ---------- ------------------------------
1.1.1.1:8080 (management)                                 test/GNMI_GNOI
         0          0 NOT CONNECTED - Dialing [1.1.1.1]:8080
```

gRPC トンネルログの収集

/volatile ディレクトリには、grpcトンネルログが格納されます。

```
bash-4.3# cd /volatile/ bash-4.3# ls /volatile -al
…
-rw-rw-rw- 1 root root 103412 Jun 21 16:14 grpc-internal-tunnel-log
…
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。