gRPC プロトコル

Google 定義されたリモート プロシージャ コール（gRPC）は、オープンソースの RPC フレームワークです。これはプロトコル バッファ（Protobuf）に基づいたオープンソースのバイナリシリアル化プロトコルです。gRPC は、XML などの構造化されたデータをシリアル化するための柔軟で効率的な自動メカニズムですが、小型で使いやすくなっています。ユーザは、.proto ファイルにプロトコル バッファ メッセージ タイプを定義することで構造を定義する必要があります。各プロトコル バッファ メッセージは、一連の名前と値のペアを含む情報の小型の論理レコードです。

Cisco gRPC インターフェイス定義言語（IDL）は、一連のサポートされている RPC（get-config、merge-config、replace-config、cli-config、delete-config、cli-show、get-models、action-json、commit、commit-replace など）を使用します。gRPC サーバは Extensible Manageability Services Daemon（EMSD）プロセス内で動作します。gRPC クライアントは任意のマシン上に配置できます。

gRPC は要求および応答をバイナリでエンコードします。gRPC は、Protobuf とともに他のコンテンツ タイプに拡張可能です。gRPC の Protobuf バイナリ データ オブジェクトは HTTP/2 を介して転送されます。

（注） gRPC を有効にする前に、TLS を設定することを推奨します。gRPC プロトコルを有効にすると、TCP で TLS が有効になっていないデフォルトの HTTP/2 トランスポートが使用されます。gRPC では、すべての gRPC 要求に対して AAA 認証および認可が義務付けられています。TLS が設定されていない場合、認証クレデンシャルは暗号化されていないネットワークを介して転送されます。非 TLS モードは、セキュアな内部ネットワークでのみ使用できます。

gRPC はクライアントとサーバ間の分散型のアプリケーションやサービスをサポートします。gRPC はサーバとクライアント間の設定データと運用データを交換するためにデバイス管理サービスを構築するインフラストラクチャを提供します。そのデータの構造は YANG モデルによって定義されます。

- サードパーティ製アプリケーションのためのトラフィック保護の制限事項 （2 ページ）
- gRPC を介したサードパーティ製アプリケーションのためのトラフィック保護の前提条件 （2 ページ）
- サードパーティ製アプリケーションのための MPP の設定 （2 ページ）

• サードパーティ製アプリケーションのためのトラフィック保護のトラブルシューティング（3 ページ）

サードパーティ製アプリケーションのためのトラフィック保護の制限事項

サードパーティ製アプリケーションのためのトラフィック保護には、以下の制限事項が適用されます。

- TPA エントリがアクティブな RP 管理インターフェイスだけを使用して設定されている場合に冗長スイッチオーバーが実行されると、gRPC 接続が失敗します。

gRPCを介したサードパーティ製アプリケーションのためのトラフィック保護の前提条件

gRPC が設定されていることを確認します。

gRPC の設定

```
Router(config)# grpc port port-number
Router(config)# grpc no-tls
Router(config-grpc)# commit
```

実行コンフィギュレーション

```
Router# show running-config grpc

grpc port 57600
 no-tls
!
```

サードパーティ製アプリケーションのためのMPPの設定

次のタスクは、サードパーティ製アプリケーションのためのトラフィック保護を設定する方法を示しています。

```
RP/0/0/CPU0:ios#configure
RP/0/0/CPU0:ios(config)#tpa
RP/0/0/CPU0:ios(config-tpa)#vrf default
RP/0/0/CPU0:ios(config-tpa-vrf)#address-family ipv4
RP/0/0/CPU0:ios(config-tpa-vrf-afi)#protection
RP/0/0/CPU0:ios(config-tpa-vrf-afi-prot)#allow protocol tcp local-port port-number
remote-address IP remote address interface interface-name local-address IP local address
```

実行コンフィギュレーション

```
Router# show running-config
tpa
vrf-default
address-family ipv4
protection
allow protocol tcp local-port 57600 remote-address 10.0.0.2/32 local-address 192.168.0.1/32
allow protocol tcp local-port 57600 remote-address 10.0.1.1/24 local-address 192.168.0.1/32
allow protocol tcp local-port 57600 remote-address 10.0.2.3/24 local-address 192.168.0.1/32


address-family ipv6
allow protocol tcp local-port 57600 remote-address 2001:DB8::1/128 local-address
2001:DB8:0:ABCD::1/128
allow protocol tcp local-port 57600 remote-address 2001:DB8::1/128 local-address
2001:DB8:0:ABCD::1/128
allow protocol tcp local-port 57600 remote-address 2001:DB8::1/128 local-address
2001:DB8:0:ABCD::1/128
   !
  !
!
```

サードパーティ製アプリケーションのためのトラフィック保護のトラブルシューティング

次の show コマンドの出力は、TPA が設定されているかどうかを確認します。

```
Router# show running-config grpc

grpc
 no-tls
!
```

次の show コマンドの出力は、TPA の設定を表示しています。

```
Router# show running-config tpa


  tpa
   vrf default
    address-family ipv4
    allow local-port 57600 protocol tcp inter mgmtEth 0/RP0/CPU0/0 local-address
192.168.0.1/32 remote-address 10.0.0.2/32
    !
```

TPA を使用しない gRPC の設定

```
Router# show kim lpts database

State:
 Prog - Programmed in hardware
 Cfg  - Configured, not yet programmed
 Ovr  - Not programmed, overridden by user configuration
 Intf - Not programmed, interface does not exist

Owner  AF Proto State       Interface      VRF          Local ip,port > Remote ip,port
------ -- ----- ----- -------------- ------------ ------------------------------
Linux   2     6         Prog                          global-vrf            any,57600
 > any,0
```

```
Router# show lpts bindings brief | include TPA
 0/RP0/CPU0 TPA  LR IPV4 TCP    default   any          any,57600 any
```

TPA を使用する gRPC の設定

次の show コマンドの出力は、LPTS データベースに設定されている内容を表示しています。また、gRPC の設定がフィルタを使用せずに Linux によって所有されているかどうかも確認します。

```
Router# show kim lpts database

State:
 Prog - Programmed in hardware
 Cfg  - Configured, not yet programmed
 Ovr  - Not programmed, overridden by user configuration
 Intf - Not programmed, interface does not exist

Owner  AF Proto State Interface      VRF          Local ip,port > Remote ip,port
------ -- ----- ----- -------------- ------------ ------------------------------
Client  2     6 Prog                 default      192.168.0.1/32,57600  > 10.0.0.2/32,0
Linux   2     6 Ovr                  global-vrf   any,57600  > any,0
```

```
Router# show lpts bindings brief | include TPA

 0/RP0/CPU0 TPA  LR IPV4 TCP    default   Mg0/RP0/CPU0/0 192.168.0.1,57600 10.0.0.2
Router#
Router# 0/RP0/ADMIN0:Mar 19 15:22:26.837 IST: pm[2433]:
%INFRA-Process_Manager-3-PROCESS_RESTART : Process tams (IID: 0) restarted
```