gRPC Tunnel

About gRPC Tunnel

Cisco NX-OS supports “gPRC Tunnel”, a specific implementation of application layer tunnel to allow external communication across network segregations like firewall.

In common network deployment, if we use firewall to roughly segregate the network to either "outside" or "inside" of the firewall. Then, for example in gNMI, the "gnmi clients (controllers)" are usually outside, while the "gnmi servers (networking switches)" are inside of the firewall. Therefore, a gNMI connection is referred as an inbound RPC to the switch, and usually referred as "Dial-In".

Such "Dial-In" results in two limitations:

- **Firewall provision:**

  "Dial-In" requires punching holes in the firewall to allow gNMI connection to go through by specifying firewall rules for specific hosts, addresses, or ports, just to name a few.

- **Prerequisite network inventory:**

  To "Dial-in" to a specific destination, an initial solicitation step is required, either manually or programmatically by the user to collect the host information of gNMI servers.

  Only after that, the client can know the address/port to "Dial-In" to those servers.

The grpc-tunnel intends to work around these two limitations, as it established tunnels using the opposite "Dial-out" approach. For more information about grpc tunnel, please refer to the external document at gRPC Tunnel

Guidelines and Limitations

The gRPC tunnel has the following guidelines and limitations:

- The naming conventions when assigning a target identifier for a tunnel is completely up to the user.

- The user is responsible to make sure the naming convention of the target identifier is unique. It is recommended that an automated deployment workflow should handle the uniqueness of the target identifier.
- Cisco NX-OS supports up to 8 tunnel configurations.

Topic 2.1

Configuring gRPC Tunnel

Configuring gRPC Tunnel without authentication

This procedure describes how to enable and configure the gRPC Tunnel without either server or client authentication. This is mostly for experimental usage.

Before you begin

gRPC Tunnel is an opaque tunnel which supposedly should be able to forward various network traffic. However, in Cisco NX-OS, the primary use case is to proxy the gNMI/gNOI requests. If that is case, it requires to properly config the grpc agent. Please refer to the gRPC Agent programming guide.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch-1# configure terminal`	Enters global configuration mode.
Step 2	**[no] feature grpctunnel** **Example:** `switch-1# feature grpctunnel`	Enables or disables grpc-tunnel feature.
Step 3	**[no] grpctunnel destination** **Example:** `switch-1# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI use-vrf management`	Configure a tunnel connection. The no form of this command removes the tunnel connection. • **destination** - (Type: IPv4/IPv6 address or hostname string) Tunnel server ip address or the hostname. If hostname is given, a valid name-server config is required. • **port** - (Type: tcp port) Tunnel server port number.

	Command or Action	Purpose
		• **target** - (Type: string, 64 bytes limit) Target ID is a string. If the user sets the ID as the reserved keyword 'HOSTNAME', the switch will substitute the switch hostname as the target. • **type** - (Type: string, 64 bytes limit) Type only supports GNMI_GNOI in 10.3.2F release. • **use-vrf** - (Type: string) The vrf name string that switch will use to dial out for grpc tunnel session. • **[Optional] source-interface** - (Type: interface name string) source-interface is used to determine the egress source ip address of the tunnel establishment. The switch would select the first ipv6 global unicast address of the interface. Else, it would select the ipv4 unicast address of the interface. This configuration supports loopback and svi interfaces only. The interface must be specified in the short name format such as Lo10, Vlan100. • **[Optional] target-vrf** - (Type: string) vrf name is used to reach local grpc server target. If not specified, uses the same as the vrf parameter. For example, specifying grpctunnel ... use-vrf foo ... target-vrf bar means the switch establishes connection to the external tunnel server in vrf foo, but forwards incoming grpc requests to the local switch grpc server residing on vrf bar.

Configuring gRPC Tunnel with Trustpoints

Note that for gRPC Tunnel, the Cisco NX-OS device initiates a connection to specified external gRPC Tunnel Destination. The user may configure the trustpoints/certificates to secure such outbound connections.

- Server authentication with “cert” option This configures the external destination’s cert into the switch. The switch would refuse to the connection if the configured cert does not match to the remote tunnel
- Client authentication with “client-cert” option This configures the identity cert of the switch. The remote tunnel server would reject the connection from the switch if the switch could not present the matching certificate.
- Mutual authentication is combination of both “cert” and “client-cert” authentication

In the below steps, step 1-3 means to import the “server authentication” while steps 4-5 mean to import the “client authentication”. The user can decide the proper combination to either enable either, or both.

Note Configuring or removing the root certificate for client authentication will cause gRPC tunnel to restart the connection to the remote destination.

Note If the client's certificate is signed by intermediate CAs, but not directly by the root CA that is imported from the above config, the grpc tunnel certification (step 5) needs to supply the full cert chain, including the user, intermediate CA cert, and the root CA cert.

Before you begin

Prepare and sign and the required certificate files for the server authentication. This is not specific to gRPC tunnel, so it is possible to re-use the existing trustpoint files.

This section means to particularly clarify the usage of the 'cert' and 'client-cert' options. The option describes the previous section can freely combine with these two options.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	(Optional) **crypto ca trustpoint** <*tunnel-trustpoint*> **Example:** `switch# crypto ca trustpoint tunnel_trustpoint`	Create a trustpoint for tunnel server authentication.
Step 3	(Optional) **rsakeypair** <*client-key*> **Example:** `switch# rsakeypar key`	Generate a rsa key pair for the client trustpoint.
Step 4	(Optional) **crypto ca authenticate** <*tunnel-trustpoint*> **Example:** `switch# crypto ca authenticate tunnel_trustpoint`	Import the tunnel server certificate. This step requires manual copy paste. Please follow the instruction.
Step 5	(Optional) **crypto ca trustpoint** <tunnel-client-trustpoint> **Example:** `switch# crypto ca trustpoint tunnel_client_trustpoint`	Create a trustpoint for tunnel client authentication.

	Command or Action	Purpose
Step 6	(Optional) **crypto ca import** <*tunnel-client-trustpoint*> **pkcs12 bootflash** *:<ca-file>* <*pkcs-password*> **Example:** `switch# crypto ca import tunnel_client_trustpoint pkcs12 bootflash:ca.pfx test`	Import the pkcs12 file to the trustpoint.
Step 7	**[no] grpctunnel destination …** **Example:** `switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI cert tunnel_trustpoint client-cert tunnel_client_trustpoint …`	Configure a tunnel connection. The no form of this command removes the tunnel connection. • **[Optional] cert** - (Type: string) Trustpoint which holds the tunnel server certificate. If not specified, would skip the server verification. • **[Optional] client-cert** - (Type: string) Trustpoint which holds the client certificate. If specified, would exercise mutual authentication with the tunnel server.

Configuring gRPC Tunnel with VRF

In the below steps, step 1-3 means to import the “server authentication” while steps 4-5 mean to import the “client authentication”. The user can decide the proper combination to either enable either, or both.

Before you begin

This section means to particularly clarify the usage of the ‘use-vrf and ‘target-vrf” options. The option describes the previous section can freely combine with these two options.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch-1# configure terminal`	Enters global configuration mode.
Step 2	**[no] feature grpctunnel** **Example:** `switch(config)# feature grpctunnel`	Enables or disables grpc-tunnel feature.
Step 3	**[no] grpctunnel destination …** **Example:**	Configure a tunnel connection. The no form of this command removes the tunnel connection.

	Command or Action	Purpose
	`switch(config)# switch(config)# grpctunnel destination 1.1.1.1 port 8000 target test1 type GNMI_GNOI use-vrf management target-vrf default`	• **use-vrf** - (Type: string) The vrf name string that switch will use to dial out for grpc tunnel session. • **[Optional] target-vrf** - (Type: string) vrf name is used to reach local grpc server target. If not specified, uses the same as the vrf parameter. For example, specifying grpctunnel ... use-vrf foo ... target-vrf bar means the switch establishes connection to the external tunnel server in vrf foo, but forwards incoming grpc requests to the local switch grpc server residing on vrf bar.

Example

This section lists a few configuration examples to illustrate the tunnel usage.

Without authentication

The following steps describe how to configure the tunnel destinations without server validation.

```
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
```

In this example, the user configures two tunnel destinations “1.1.1.1:8000” and “server.foo.com:8000” with target “test1” and “test2” respectively. The connections are initiated over the management namespace.

With server authentication

The following steps describe how to configure the tunnel destinations with server validation.

Execute the following commands to Import server cert to the trustpoint.

```
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
```

Then execute the following command to configure the tunnel destination.

Also use the show command to display the configuration.

```
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"
}
}
}
]
}
}
]
}
}
```

With client authentication

The following steps describe how to configure the tunnel destinations with client validation.

The following steps describe how to configure the tunnel destinations without server validation.

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

Then execute the following command to configure the tunnel destination.

Also use the show command to display the configuration.

```
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"
}
}
}
]
}
}
]
}
}
```

With VRF

The combination of 'use-vrf' and 'target-vrf' config offers deployment flexibility but may also incur confusions.

Please note the following difference.

- use-vrf à How to reach the remote tunnel destination.
- target-vrf à How to forward/relay the tunnel traffic to the switch’s internal service.

Please refer to the below example scenarios:

- The remote tunnel server is reachable via the 'management' vrf. When the switch received a gNMI connection within the tunnel, the switch would forward to the gnmi 'management' server.

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

- The remote tunnel server is reachable via the 'management' vrf, while the local grpc agent is running on the default vrf. With the below config, When the switch received a gNMI connection within the tunnel, the switch would stich the gnmi request to the to default vrf.

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

- Both the remote tunnel server and the local grpc agent are running on the default vrf.

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

- The remote tunnel server is reachable via the 'default' vrf, while the local grpc agent is running on the ‘test” vrf. With the below config, When the switch received a gNMI connection within the tunnel, the switch would stich the gnmi request to the to test vrf.

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

- In this case, the remote tunnel server is reachable via the 'default' vrf, while the local grpc agent is running on the ‘abc” vrf. With the below config, When the switch received a gNMI connection within the tunnel, the switch would stich the gnmi request to the to test vrf, thus the connection would not work. This can be treated as a forward reference. The connection would start to work after changing grpc config to 'grpc use-vrf test'.

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

Troubleshooting

Check Feature Status

- In Cisco NX-OS, enter the **show feature grpctunnel** command to check the agent config.
- To view the status of the gRPC tunnel, use the **show feature** command.

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

Debug gRPC Tunnel

There exist a series of show commands to display tunnel status.

Show Commands

To verify the tunnel configuration/status, enter the following command:

Command	Description

show grpctunnel internal sessions [all] { summary | detail } }	Displays the tunnel status. • The 'sessions' option would list the tu The 'all' option would list the ended sessio For ended sessions, the switch would retai
debug grpctunnel events all	Executed in CLI EXEC mode This allows to display the debug message

Example Output

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
```

Gathering gRPC Tunnel Logs

The /volatile directory houses the grpc tunnel log

```
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
…
```