Configure Policy-based Telemetry

Policy-based telemetry (PBT) streams telemetry data to a destination using a policy file. A policy file defines the data to be streamed and the frequency at which the data is to be streamed.

ASR9000 series routers and CRS routers do not support PBT.

The process of streaming telemetry data uses three core components:

- **Telemetry Policy File** specifies the kind of telemetry data to be generated, at a specified frequency.
- **Telemetry Encoder** encapsulates the generated data into the desired format and transmits to the receiver.
- **Telemetry Receiver** is the remote management system that stores the telemetry data.

For more information about the three core components, see Core Components of Policy-based Telemetry Streaming.

Note Model-driven telemetry supersedes policy-based telemetry.

Streaming policy-based telemetry data to the intended receiver involves these tasks:

Create Policy File

You define a telemetry policy file to specify the kind of telemetry data to be generated and pushed to the receiver. Defining the policy files requires a path to stream data. The paths can be schemas, native YANG or allowed list entries.

For more information on the schema paths associated with a corresponding CLI command, see Schema Paths.

For more information on policy files, see Telemetry Policy File.

1. Determine the schema paths to stream data.

   For example, the schema path for interfaces is:

```
RootOper.InfraStatistics.Interface(*).Latest.GenericCounters
```

2. Create a policy file that contains these paths.

Example: Policy File

The following example shows a sample policy file for streaming the generic counters of an interface:

```
{
"Name": "Test",
 "Metadata": {
     "Version": 25,
     "Description": "This is a sample policy",
     "Comment": "This is the first draft",
     "Identifier": "<data that may be sent by the encoder to the mgmt stn"
},
 "CollectionGroups": {
     "FirstGroup": {
         "Period": 10,
         "Paths": [
             "RootOper.InfraStatistics.Interface(*).Latest.GenericCounters"
             ]
     }
 }
}
```

The following example shows the paths with allowed list entries in the policy file. Instead of streaming all the data for a particular entry, only specific items can be streamed using allowed list entries. The entries are allowed using `IncludeFields` in the policy file. In the example, the entry within the IncludeFields section streams only the latest applied AutoBW value for that TE tunnel, which is nested two levels down from the top level of the path:

```
{

        "Name": "RSVPTEPolicy",
        "Metadata": {

                "Version": 1,
                "Description": "This policy collects auto bw stats",
                "Comment": "This is the first draft"
        },

        "CollectionGroups": {
                "FirstGroup": {
                   "Period": 10,
                        "Paths": {
                            "RootOper.MPLS_TE.P2P_P2MPTunnel.TunnelHead({'TunnelName':
'tunnel-te10'})": {
                                "IncludeFields": [{

                                     "P2PInfo": [{
                                         "AutoBandwidthOper": [
                                               "LastBandwidthApplied"
                                               ]
                                         }]
                                    }]
                                }
                        }
                }
```

```
        }
}
```

The following example shows the paths with native YANG entry in the policy file. This entry will stream the generic counters of the interface:

```
"Paths": [

"/Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface=*/latest/generic-counters"

 ]
```

What to Do Next:

Copy the policy file to the router. You may copy the same policy file to multiple routers.

Copy Policy File

Run the Secure Copy Protocol (SCP) command to securely copy the policy file from the server where it is created. For example:

```
$ scp Test.policy <ip-address-of-router>:/telemetry/policies
```

For example, to copy the `Test.policy` file to the `/telemetry/policies` folder of a router with IP address 10.0.0.1:

```
$ scp Test.policy cisco@10.0.0.1:/telemetry/policies
cisco@10.0.0.1's password:
Test.policy
100%  779     0.8KB/s   00:00
Connection to 10.0.0.1 closed by remote host.
```

Verify Policy Installation

In this example, the policy is installed in the `/telemetry/policies/` folder in the router file system. Run the **show telemetry policies brief** command to verify that the policy is successfully copied to the router.

```
Router#show telemetry policy-driven policies brief
Wed Aug 26 02:24:40.556 PDT

 Name                        |Active?| Version | Description
-----------------------------|-------|---------|------------------------------
 Test                          N        1        This is a sample policy
```

What to Do Next:

Configure the telemetry encoder to activate and stream data.

Configure Encoder

An encoder calls the streaming Telemetry API to:

- Specify policies to be explicitly defined
- Register all policies of interest

Configure the encoder to activate the policy and stream data. More than one policy and destination can be specified. Multiple policy groups can be specified under each encoder and each group can be streamed to multiple destinations. When multiple destinations are specified, the data is streamed to all destinations.

Configure an encoder based on the requirement.

Configure JSON Encoder

The JavaScript Object Notation (JSON) encoder is packaged with the IOS XR software and provides the default format for streaming telemetry data.

To stream data in JavaScript Object Notation (JSON) format, specify the encoder, policies, policy group, destination, and port:

```
Router# configure
Router(config)#telemetry policy-driven encoder json
Router(config-telemetry-json)#policy group FirstGroup
Router(config-policy-group)#policy Test
Router(config-policy-group)#destination ipv4 10.0.0.1 port 5555
Router(config-policy-group)#commit
```

The names of the policy and the policy group must be identical to the policy and its definition that you create. For more information on policy files, see Create Policy File, on page 1.

For more information about the message format of JSON encoder, see JSON Message Format

Configure GPB Encoder

Configuring the GPB (Google Protocol Buffer) encoder requires metadata in the form of compiled `.proto` files. A `.proto` file describes the GPB message format, which is used to stream data.

Two encoding formats are supported:

- **Compact encoding** stores data in a compressed and non-self-describing format. A `.proto` file must be generated for each path in the policy file to be used by the receiver to decode the resulting data.
- **Key-value encoding** uses a single `.proto` file to encode data in a self-describing format. This encoding does not require a `.proto` file for each path. The data on the wire is much larger because key names are included.

To stream GPB data, complete these steps:

1. For compact encoding, create `.proto` files for all paths that are to be streamed using the following tool:

   ```
   telemetry generate gpb-encoding path <path> [file <output_file>]
   ```

   or

   ```
   telemetry generate gpb-encoding policy <policy_file> directory <output_dir>
   ```

   Attention A parser limitation does not support the use of quotes within paths in the tool. For example, for use in the tool, change this policy path, `RootOper.InfraStatistics.Interface(*).Latest.Protocol(['IPV4_UNICAST'])` to `RootOper.InfraStatistics.Interface(*).Latest.Protocol`.

2. Copy the policy file to the router.
3. Configure the telemetry policy specifying the encoder, policies, policy group, destination, and port:

```
Router# configure
Router(config)#telemetry policy-driven encoder gpb
Router(config-telemetry-json)#policy group FirstGroup
Router(config-policy-group)#policy Test
Router(config-policy-group)#destination ipv4 10.0.0.1 port 5555
Router(config-policy-group)#commit
```

For more information about the message format of GPB encoder, see GPB Message Format

Verify Policy Activation

Verify that the policy is activated using the `show telemetry policies` command.

```
Router#show telemetry policy-driven policies
Wed Aug 26 02:24:40.556 PDT

  Filename:            Test.policy
  Version:              25
  Description:          This is a sample policy to demonstrate the syntax
  Status:               Active
  CollectionGroup: FirstGroup
    Cadence:              10s
    Total collections:    2766
    Latest collection:    2015-08-26 02:25:07
    Min collection time:  0.000s
    Max collection time:  0.095s
    Avg collection time:  0.000s
    Min total time:       0.022s
    Max total time:       0.903s
    Avg total time:       0.161s
    Collection errors:    0
    Missed collections:   0

+-----------------------------------------+--------+--------+------+------+------+------+------+------+

    | Path                                                                  | Avg (s)
 | Max (s) | Err  |

+-----------------------------------------+--------+--------+------+------+------+------+------+------+

    | RootOper.InfraStatistics.Interface(*).Latest.GenericCounters             |   0.000 |
   0.000 |    0 |

+-----------------------------------------+--------+--------+------+------+------+------+------+------+
```

After the policy is validated, the telemetry encoder starts streaming data to the receiver. For more information on the receiver, see Telemetry Receiver.