Splunk HEC Exporter Change Proposal for Health Metrics

Background

As part of DMX-10614 we want to add metrics indicating exporter health. This in turn will be consumed by UI to provide actionable alerts to the Edge Processor (EP) users, initially, the most obvious errors at the exporter. The metric spec is as follows and suggests the following health error classes to classify errors at the exporter:

```
Unset
edge_processor_export_error_count
"""Shows the number of errors that have occurred when exporting data
to a data destination"""

Standard dimensions: (instance, destinationId)

Dimension: errorType
Values:
1. HostNotFound
2. ConnectionForbidden
3. NotAuthenticated
4. ResourceNotFound
5. NotAuthorized
6. InvalidRequest
7. Unclassified
```

Goal

Our goal is for the users of OTEL collector HEC exporter to be able to emit metrics based on different error types. So they can understand the health information subsequently.

Approach

Since Splunk HEC exporter is part of the opentelemetry-collector-contrib OSS repo, we would like to propose two plausible options for implementation that can be generally applicable to a larger audience. At a high level we propose following two options for the change in Splunk HEC exporter:

1. Add metrics on HTTP call with customizable metrics name and dimensions
2. Allow wrapping of `push[Metrics|Log]Data` method

Option 1: Add metrics on HTTP call with customizable metrics name and dimensions

In this scenario, we propose adding health metrics surrounding the HTTP call. To make this change generally applicable and to not be specific to edge processor metric, i.e., `edge_processor_export_error_count`, we can allow for customizable metric name and allow for customizable dimension name (`errorType`) and its values for HTTP status codes by reusing HecTelemetry config.

Pros:

- Good enhancement to the exporter so OSS can re-use
- Ensure compatibility of the metrics as the exporter evolves over time

Cons:

- More intrusive
- More and more metrics around exporter so it can become hard to maintain

Option 2: Allow wrapping of push[Metrics|Log]Data method

In this scenario, we propose exposing `pushMetricsData` and `pushLogData` to allow devs to wrap Splunk HEC Exporter into their project. This is not possible today because of how these two functions are wrapped in the exporterhelper before they are attached to ConsumeMetrics or ConsumeLogs functions. This prevents the users of this exporter from receiving HTTP status code errors since the consumer retries forever and never returns to the wrapped function.

Pros:

- Increased customization. Users of the code can customize what they want to do with the errors
- Less bloated metrics

Cons:

- Wrapper is generally not configurable via OTEL config. This disobeys some of the OTEL principles and creates a risk for incompatibility as the exporter API can change over time.