NETCONF
Internet-Draft
Intended status: Standards Track
Expires: 9 January 2025

R. Gagliano
Cisco Systems
K. Larsson
Deutsche Telekom AG
J. Lindblad
Cisco Systems

8 July 2024

RESTCONF Extension to ~~support~~ Support Trace Context Headers

draft-ietf-netconf-restconf-trace-ctx-headers-01

Abstract

This document ~~extends~~ defines an extension to the RESTCONF protocol in order to support trace
context propagation as defined by the W3C.

Commenté [BMI1]: To avoid this to be misinterpreted as we updated RFC8040.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://github.com/netconf-wg/restconf-trace-ctx-headers/blob/gh-pages/draft-ietf-netconf-restconf-trace-ctx-headers.txt. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-netconf-restconf-trace-ctx-headers/.

Discussion of this document takes place on the NETCONF Working Group mailing list (mailto:netconf@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/netconf/. Subscribe at https://www.ietf.org/mailman/listinfo/netconf/.

Source for this draft and an issue tracker can be found at https://github.com/netconf-wg/restconf-trace-ctx-headers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 9 January 2025.

Copyright Notice

Table of Contents

1. Introduction

Network ~~automation and~~ management (including automation) systems commonly consist of multiple sub-systems and, together with the network devices they manage, they effectively form a distributed system. Distributed tracing is a methodology implemented by tracing tools to ~~follow~~track, analyze, and debug operations~~,~~ such as configuration transactions, across multiple distributed systems. In such context, ~~A~~an operation is uniquely identified by an identifier (called ~~-~~trace-id) and through a trace context, carries some metadata about the operation. Propagating this "trace context" between systems ~~enables~~ is meant to provide ~~forming~~ a coherent view of the entire operation as carried out by all involved systems.

The W3C has defined two HTTP headers (_traceparent_ and _tracestate_) for context propagation that are useful for distributed systems like the ones defined in [RFC8309]. The goal of this document is to adopt

Commenté [BMI2]: Please add an authoritative pointer.

Commenté [BMI3]: Not sure which part you are referring to.

I suspect Section 4 (with is only about an example). If so, please add an explicit pointer to that section.

this W3C specification for the RESTCONF protocol.

This document does not define new HTTP extensions but makes those defined in [W3C-Trace-Context] optional headers for the RESTCONF protocol [RFC8040].

~~In~~ [I-D.~~draft-~~ietf-netconf-trace-ctx-extension~~-01~~] defines a~~, the~~ NETCONF protocol extension for trace purposes. ~~is defined and we will~~ The present document ~~re-use~~leverages ~~-~~several of the YANG and XML objects defined in that document~~ for RESTCONF~~. Readers should~~Please~~ refer to [I-D.ietf-netconf-trace-ctx-extension] ~~that document~~ for additional context and example applications.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT","SHOULD","SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. RESTCONF Extensions

A RESTCONF server MUST support the trace context _traceparent_ header as defined in [W3C-Trace-Context].

A RESTCONF server SHOULD support the trace context _tracestate_ header as defined in [W3C-Trace-Context].

2.1. Error Handling

~~The~~ A RESTCONF server SHOULD follow the "Processing Model for Working with Trace Context" as specified in [W3C-Trace-Context]. Based on this processing model, it is NOT RECOMMENDED to reject an RPC because of the trace context header values.

If ~~the~~ a server ~~still~~ decides to reject ~~the~~ an RPC because of the trace context header values, the server MUST return a RESTCONF rpc-error with the following values:

```
error-tag:      operation-failed
error-type:     protocol
error-severity: error
```

Additionally, the error-info ~~tag~~ MUST contain relevant details about the error in the form of an sx:structure otlp-trace-context-error-info defined in "ietf-netconf-otlp-context~~.yang~~" module ~~from~~ [I-D.~~draft-~~ietf-netconf-trace-ctx-extension~~-01~~].

2.2. Trace Context header V~~v~~ersionning

This extension refers to the [W3C-Trace-Context] trace context capability. The W3C _traceparent_ and _tracestate_ headers include the notion of versions. It ~~would be~~is desirable for a RESTCONF client to ~~be able to~~ discover the one or multiple versions of these headers

Commenté [BMI4]: How is this optional then?

Commenté [BMI5]: Can we call exception case where this SHOULD can be safely ignored?

Commenté [BMI6]: The W3C spec says "*This section is non-normative.*"

Commenté [BMI7]: Isn't this deployment-specific/policy-based?

Commenté [BMI8]: Should we define an error code for cases where the response trace values do not match the ones in the request? Actually, shouldn't that behavior be called out here?

Commenté [BMI9]: Which one?

supported by a server. We would like to achieve this goal avoiding the definition of new RESTCONF capabilities for each headers' version.

[I-D.~~draft-~~ietf-netconf-trace-ctx-extension~~-01~~] defines a pair of YANG modules that MUST be included in the YANG library per [RFC8525] of the RESTCONF server supporting the RESTCONF Trace Context extension that will refer to the headers' supported versions. Future updates of this document could include additional YANG modules for new headers' versions.

3. Security Considerations

There are two YANG modules specified in this document. These modules are completely empty, and therefore have very limited security considerations. Their purpose is only to indicate which trace context header versions the server supports using YANG Library [RFC8525].

~~Even though both YANG modules are empty, there are still some security considerations worth mentioning, however. This is because the functionality described in this document is in the form of additional HTTP headers (which cannot be described using YANG) relating to the network management protocol RESTCONF [RFC8040].~~

The _traceparent_ and _tracestate_ headers make it easier to track and correlate the flow of requests and their downstream effect on other systems. This is indeed the whole point with these headers. This knowledge ~~could also be of~~may be used ~~to by bad actors~~unauthorized entities ~~that are working to build~~to infer a map of ~~the~~ a managed network.

All advice mentioned in the [W3C-Trace-Context] under the Privacy Considerations and Security Considerations also apply to this document.

The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

~~The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.~~

4. IANA Considerations

This document has no IANA actions.

5. Acknowledgments

The authors would like to acknowledge the valuable implementation feedback from Christian Rennerskog and Per Andersson. Many thanks to Raul Rivas Felix, Alexander Stoklasa, Luca Relandini and Erwin Vrolijk for their help with the demos regarding integrations. The help and support from Jean Quilbeuf and Benoît Claise has also been invaluable to this work.

Commenté [BMI10]: Not sure I would keep this as such. I would use a more formal language to describe the behavior.

Commenté [BMI11]: ?

Commenté [BMI12]: No modules are defined in this document.

Commenté [BMI13]: Not sure I would maintain this.

6. References

6.1. Normative References

[I-D.draft-ietf-netconf-trace-ctx-extension-01]
Gagliano, R., Larsson, K., and J. Lindblad, "NETCONF Extension to support Trace Context propagation", Work in Progress, Internet-Draft, draft-ietf-netconf-trace-ctx-extension-01, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-trace-ctx-extension-01>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/rfc/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/rfc/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/rfc/rfc8525>.

[W3C-Trace-Context]
"W3C Recommendation on Trace Context", 23 November 2021, <https://www.w3.org/TR/2021/REC-trace-context-1-20211123/>.

6.2. Informative References

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/rfc/rfc8309>.

Appendix A. Example RESTCONF Cealls

All examples from ~~[RFC8040]~~ Appendix B of [RFC8040] could be recreated in this ~~seciton~~ section by adding the new header described in this document. We selected one example from that document as reference.

A.1. Successful creation New Data Resources (from ~~section~~ Appendix B.2.1 ~~in~~of
[RFC8040])

To create a new "artist" resource within the "library" resource, ~~the~~a client might send the following request:

```
POST /restconf/data/example-jukebox:jukebox/library HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
traceparent: 00-405062f633be64ee006089dfca95a153-e021f9e263aad8e2-01
tracestate: vendorname1=opaqueValue1,vendorname2=opaqueValue2

{
  "example-jukebox:artist" : [
    {
      "name" : "Foo Fighters"
    }
  ]
}
```

If the resource is created, the server might respond as follows:

```
HTTP/1.1 201 Created
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Location: https://example.com/restconf/data/\
    example-jukebox:jukebox/library/artist=Foo%20Fighters
Last-Modified: Thu, 26 Jan 2017 20:56:30 GMT
ETag: "b3830f23a4c"
traceparent: 00-405062f633be64ee006089dfca95a153-e021f9e263aad8e2-01
tracestate: vendorname1=opaqueValue1,vendorname2=opaqueValue2
```

A.2. Unsuccessful ~~creation~~ Creation New Data Resources (from Appendix~~section~~ B.2.1 ~~in~~of
[RFC8040])

[W3C-Trace-Context] specifies that vendor ~~MAY~~ may validate the _tracestate_ header and that invalid headers ~~MAY~~ may be discarded. ~~In the section about Error handling (~~Section 2.1~~), it is stated~~ states that servers ~~MAY~~ may return an error. Let's assume that ~~is~~ an ~~our~~ implementation follow that behavior.

Example of a badly formated _tracestate_ header using [RFC8040] example (Appendix B.2.1), which by following~~:~~

```
POST /restconf/data/example-jukebox:jukebox/library HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
traceparent: 00-405062f633be64ee006089dfca95a153-e021f9e263aad8e2-01
tracestate: SomeBadFormatHere

{
  "example-jukebox:artist" : [
    {
      "name" : "Foo Fighters"
```

Commenté [BMI14]: To be udpated

Commenté [BMI15]: Inappropriate use of normative language.

Commenté [BMI16]: Please double check.

Commenté [BMI17]: I don't parse this.

```
      }
    ]
  }
```

And the expected error message:

```
 HTTP/1.1 400 Bad Request
 Date: Tue, 20 Jun 2023 20:56:30 GMT
 Server: example-server
 Content-Type: application/yang-data+json

 { "ietf-restconf:errors" : {
     "error" : [
       {
         "error-type" : "protocol",
         "error-tag" : "operation-failed",
         "error-severity" : "error",
         "error-message" :
           "OTLP traceparent attribute incorrectly formatted",
         "error-info": {
           "ietf-netconf-otlp-context:meta-name" : "tracestate",
           "ietf-netconf-otlp-context:meta-value" :
             "SomeBadFormatHere",
           "ietf-netconf-otlp-context:error-type" :
             "ietf-netconf-otlp-context:bad-format"
         }
       }
     ]
   }
 }
```

Commenté [BMI18]: To be updated to use a more recent date