Network Working Group
Internet-Draft
Intended status: Historic
Expires: August 14, 2012

R. Gerhards
Adiscon GmbH
C. Lonvick
Cisco Systems, Inc.

February 11, 2012

Transmission of Syslog Messages over TCP
draft-gerhards-syslog-plain-tcp-14.txt

Abstract

There have been many implementations and deployments of legacy syslog over TCP for many years. That protocol has evolved without being standardized and has proven to be quite interoperable in practice. This memo describes how TCP has been used as a transport for syslog messages.

Status of this Memo

Copyright Notice

Table of Contents

1. Statement by the IESG

The IESG does not recommend implementing or deploying syslog over plain tcp, which is described in this document, because it lacks the ability to enable strong security [RFC3365].

The TLS transport [RFC5425] is recommended for implementation so that appropriate security features are available to operators who want to deploy secure syslog. Similarly, those security features can be turned off for those who do not want them.

2. Introduction

The standards track documents in the syslog series recommend using the syslog protocol [RFC5424] with the TLS transport [RFC5425] for all event messages. The authors of this document wholeheartedly support that position and only offer this document to describe what has been observed with legacy syslog over TCP, which appears to still be widely used.

Two primary format options have been observed with legacy syslog being transported over TCP. These have been called non-transparent-framing and octet-counting. The non-transparent-framing mechanism has some inherent problems.

Diagram 1 shows how all of these syslog transports relate to each other. In this diagram three originators are seen, labeled A, B, and C, along with one collector. Originator A is using the TCP transport which is described in this document. Originator B is using the UDP transport, which is described in [RFC5426]. Originator C is using the TLS transport, which is described in [RFC5425]. The collector is shown with the capability to accept all three transports.

```
    +---------------------+
    | Originator A        |
    |---------------------|
    |  syslog application |
    |                     |
    |---------------------|
    |  syslog transport   |
    |        TCP          |
    |---------------------|
              v
              |
             /                            +---------------------+
            /                             | Originator B        |
           /                              |---------------------|
          /   +----------------------+    |  syslog application |
         /    | Collector            |    |                     |
        |     |----------------------|    |---------------------|
        |     |  syslog application  |    |  syslog transport   |
        |     |                      |    |        UDP          |
        |     |----------------------|    |---------------------|
        |     |  syslog transport    |              v
        |     |  TCP |  TLS  |  UDP  |              |
        |     |----------------------|              |
        |         ^      ^       ^                  |
        |         |      |       |                  |
        \         /      |       \                  /
         ---------       |        ------------------
                         |
                         |
                         |     +---------------------+
                         |     | Originator C        |
                         |     |---------------------|
                         |     |  syslog application |
                         |     |                     |
                         |     |---------------------|
                         |     |  syslog transport   |
                         |     |        TLS          |
                         |     |---------------------|
                         |               v
                         \               /
                          ---------------
```

Diagram 1. Syslog Layers

3. Conventions Used in This Document

The terminology defined in Section 3 of [RFC5424] is used throughout this specification. The reader should be familiar with that to follow this discussion.

This document also references devices that use the syslog message format as described in [RFC3164]. Devices that continue to use that message format (regardless of transport) will be described as "legacy syslog devices". Similarly, devices that use the message format as described in [RFC5424] will be described as "standardized syslog devices".

4. Message Transmission

Syslog is simplex in nature. It has been observed that implementations of syslog over TCP also do not use any backchannel mechanism to convey information to the transport sender, and consequently do not use any application-level acknowledgement for syslog receiver to sender signaling. Message receipt acknowledgement, reliability, and flow control are provided by the capabilities of TCP.

4.1. Character Encoding Scheme

Syslog over TCP messages contain no indication of the coded character set (e.g., [US-ASCII] or [UNICODE]) or character encoding scheme (e.g., so-called "7-bit ASCII" or UTF-8 [RFC3629]) in use. In these messages, the predominant approach has been to include characters only from the ASCII repertoire (i.e., %d32 to %d126 inclusive) using the "Network Virtual Terminal" (NVT) encoding [RFC5198].

The message header usually contains characters only from the ASCII repertoire, in the NVT encoding. This has been observed even in cases where a different encoding (e.g., UTF-8) has been used for the MSG part. However, characters outside the ASCII range have been seen inside the header. In that case, some syslog applications have been known to experience problems processing those messages.

In some cases, it has been observed that characters outside of the ASCII range are often being transformed by receivers in an effort to "escape control characters". Some receiver implementations simply drop those characters. This is considered to be a poor practice as it causes problems with coded character sets other than ASCII and character encodings other than NVT, most notably the UTF-8 encoding of Unicode.

It has also been observed that relays will forward messages using the character encoding schemes of messages they receive. In the case where two different senders are using different character encoding schemes, the relay will forward each message to a collector in that character encoding. The collector of these messages will have to be prepared to receive messages from the same relay with different encodings.

4.2. Session

Like most other protocols, the syslog transport sender is the TCP host that initiates the TCP session. After initiation, messages are sent from the transport sender to the transport receiver. No application-level data is transmitted from the transport receiver to the transport sender. The roles of transport sender and receiver seem to be fixed once the session is established.

When it has been observed, if an error occurs that cannot be corrected by TCP, the host detecting the error gracefully closes the TCP session. There have been no application level messages seen that were sent to notify the other host about the state of the host syslog application.

4.3. Session Initiation

The TCP host acting as a syslog transport receiver listens to a TCP port. The TCP transport sender initiates a TCP session to the syslog transport receiver as specified in [RFC0793].

This protocol has no standardized port assignment. In practice, network administrators generally choose something that they feel will not conflict with anything else active in their networks. This has most often been either TCP/514, which is actually allocated to another protocol, or some variant of adding 514 to a multiple of 1000. Please see Section 5 for more information about this.

4.4. Message Transfer

Syslog over TCP has been around for a number of years. Just like legacy syslog over UDP, different implementations exist. The older method of non-transparent-framing has problems. The newer method of octet-counting is reliable and has not been seen to cause problems noted with the non-transparent-framing method.

In both of these methods, during the message transfer phase, the syslog transport sender sends a stream of messages to the transport receiver. These are sent in sequence and one message is encapsulated inside each TCP frame. Either of the TCP hosts may initiate session

closure at any time as specified in Section 3.5 of [RFC0793]. In practice, this is often seen after a prolonged period of inactivity.

4.4.1. Octet Counting

This framing allows for the transmission of all characters inside a syslog message and is similar to the method used in [RFC5425]. A transport receiver uses the defined message length to delimit a syslog message. As noted in [RFC3164] the upper limit for a legacy syslog message length is 1024 octets. That length has been expanded for standardized syslog.

It can be assumed that octet-counting framing is used if a syslog frame starts with a digit.

All syslog messages can be considered to be TCP "data" as per Transmission Control Protocol [RFC0793]. The syslog message stream has the following ABNF [RFC5234] definition:

```
    TCP-DATA = *SYSLOG-FRAME

    SYSLOG-FRAME = MSG-LEN SP SYSLOG-MSG   ; Octet-counting
                                           ; method

    MSG-LEN = NONZERO-DIGIT *DIGIT

    NONZERO-DIGIT = %d49-57

    SYSLOG-MSG is defined in the syslog protocol [RFC5424] and may
               also be considered to be the payload in [RFC3164]
```

MSG-LEN is the octet count of the SYSLOG-MSG in the SYSLOG-FRAME.

4.4.2. Non-Transparent-Framing

The non-transparent-framing method inserts a syslog message into a frame and terminates it with a TRAILER character. The TRAILER has usually been a single character and most often is ASCII LF (%d10). However, other characters have also been seen, with ASCII NUL (%d00) being a prominent example. Some devices have also been seen to emit a two-character TRAILER, which is usually CR and LF.

The problem with non-transparent-framing comes from the use of a TRAILER character. In that, the traditional trailer character is not escaped within the message, which causes problems for the receiver. For example, a message in the style of [RFC3164] containing one or more LF characters may be misinterpreted as multiple messages by the

receiving syslog application.

The ABNF for this is shown here:

```
    TCP-DATA = *SYSLOG-FRAME

    SYSLOG-FRAME = SYSLOG-MSG TRAILER  ; non-transparent-framing
                                       ; method

    TRAILER = LF / APP-DEFINED

    APP-DEFINED = 1*2OCTET

    SYSLOG-MSG is defined in the syslog protocol [RFC5424] and may
               also be considered to be the payload in [RFC3164]
```

A transport receiver can assume that non-transparent-framing is used if a syslog frame starts with the ASCII character "<" (%d60).

4.4.3. Method Change

It has been observed in legacy implementations that the framing may change on a frame-by-frame basis. This is probably not a good idea, but it's been seen.

4.5. Session Closure

The syslog session is closed when one of the TCP hosts decides to do so. It then initiates a local TCP session closure. Following TCP [RFC0793] it doesn't need to notify the remote TCP host of its intention to close the session, nor does it accept any messages that are still in transit.

5. Applicability Statement

Again it must be emphasized that the standards track documents in the syslog series recommend using the TLS transport [RFC5425] to transport syslog messages. This document does not recommend that new implementations or deployments use syslog over TCP except for the explicit purpose of interoperating with existing deployments.

One of the major problems with interoperability with this protocol is that there is no consistent TCP port assigned. Most of the successful implementations have made the selection of a port a user-configurable option. The most frequently observed port for this has been TCP/514, which is actually assigned to the Shell protocol.

Operators must carefully select which port to use in their deployment and be prepared to encounter different default port assignments in implementations.

There are several advantages to using TCP: flow control, error recovery, and reliability, to name a few. These reasons and the ease of programming have lead people to use this transmission protocol to transmit syslog.

One potential disadvantage is the buffering mechanism used by TCP. Ordinarily, TCP decides when enough data has been received from the application to form a segment for transmission. This may be adjusted through timers but still, some application data may wait in a buffer for a relatively long time. Syslog data is not normally time-sensitive but if this delay is a concern, the syslog transport sender may utilize the PUSH Flag as described in [RFC0793] to have the sending TCP immediately send all buffered data.

6. Security Considerations

This protocol makes no meaningful provisions for security. It lacks authentication, integrity checking, and privacy. It makes no provision for flow control or end-to-end confirmation of receipt, relying instead on the underlying TCP implementations to approximate these functions. It should not be used if deployment of [RFC5425] on the systems in question is feasible.

7. IANA Considerations

There are no requests for IANA actions in this document.

8. Acknowledgments

The authors wish to thank David Harrington, Tom Petch, Richard Graveman, and all other people who commented on various versions of this proposal. We would also like to thank Peter Saint-Andre for clarifying character encodings.

The authors would also like to thank Randy Presuhn for being our reviewer and document shepherd, and a special thanks to Dan Romascanu for his support and guidance.

9. Notes to the RFC Editor and Change Log

These are notes to the RFC editor. Please delete this section after the notes have been followed.

Version -14 addresses the final few IESG requests. It was submitted in February of 2012.

Version -13 addressed the IESG reviews and is changed to Historic. It was submitted in January of 2012.

Version -12 addressed AD Review comments as well as GENART comments. It was submitted in December of 2011.

Version -11 fixed the ABNF and was submitted in October of 2011.

Version -10 was put together based on Randy Presuhn's feedback as shepherd. A section on character sets has been added. The term "octet-stuffing" was incorrectly used and has been replaced by "non-transparent-framing". The security considerations section has been simplified. It was submitted in October of 2011.

Version -09 was put together based on IESG member feedback. The appendixes were removed and things were consolidated to be more appropriate for an informational document. It was submitted in August of 2011. Dan Romascanu is actually the IESG member who will watch this document.

Version -08 included a reference to vulnerabilities of TCP. It was submitted in February of 2011.

Version -07 was submitted in January, 2011. This clarified what was really expected from what was optional. Appendix B was added for further clarification. Additionally, the security Considerations section was edited to include a discussion about transport layer issues.

Version -06 was submitted in October, 2010. The 2119 language was removed. Also, we compared notes and couldn't find any implementations that stacked multiple messages in a frame in the octet-counting method. That paragraph was removed.

Version -05 was submitted in September, 2010 to address some items that David Harrington noted as he is becoming the document shepherd.

Version -04 was submitted in April, 2010 to clean up some items.

Version -03 was submitted in April, 2010 based upon further review

comments from Tom Petch.

Version -02 was submitted in March, 2010 based upon review comments from Tom Petch.

Version -01 was submitted based upon review comments from David Harrington.

Version -00 was created in November, 2009.

10. References

10.1. Normative

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, August 2002.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

[RFC5425] Miao, F., Ma, Y., and J. Salowey, "Transport Layer Security (TLS) Transport Mapping for Syslog", RFC 5425, March 2009.

[US-ASCII] ANSI, "Coded Character Set -- 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986", 1968.

10.2. Informative

[RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164, August 2001.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5426] Okmianski, A., "Transmission of Syslog Messages over UDP", RFC 5426, March 2009.

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 6.0"", 2010, <http://www.unicode.org/versions/Unicode6.0.0/>.

Authors' Addresses

Rainer Gerhards
Adiscon GmbH
Mozartstrasse 21
Grossrinderfeld, BW 97950
Germany

Email: rgerhards@adiscon.com

Chris Lonvick
Cisco Systems, Inc.
12515 Research Blvd.
Austin, TX 78759
USA

Email: clonvick@cisco.com