Transmission of Syslog Messages over TCP
draft-gerhards-syslog-plain-tcp-03.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.

The aim of this specification is to document three things: how to transmit standardized syslog over TCP, how this has been done for legacy syslog, and how the new syslog architecture can interoperate with the legacy deployments.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 3, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.
Table of Contents

1.  Introduction ............................................. 3
2.  Conventions Used in This Document ........................ 5
3.  Message Transmission .................................... 5
   3.1.  Session .............................................. 5
   3.2.  Session Initiation .................................. 6
   3.3.  Message Transfer ..................................... 6
      3.3.1.  Octet-Counting ................................. 7
      3.3.2.  Octet-Stuffing ................................ 7
   3.4.  Retaining the Original Message ....................... 8
   3.5.  Session Closure ..................................... 8
4.  Security Considerations .................................. 8
   4.1.  Sender Authentication and Message Forgery ............ 8
   4.2.  Message Observation ................................ 9
   4.3.  Replaying .......................................... 9
   4.4.  Message Prioritization and Differentiation .......... 10
   4.5.  Denial of Service .................................. 10
   4.6.  Reliability ........................................ 10
5.  IANA Considerations .................................... 10
6.  Acknowledgments ........................................ 10
7.  Notes to the RFC Editor and Change Log .................. 10
8.  References ............................................. 11
   8.1.  Normative .......................................... 11
   8.2.  Informative ....................................... 11
Appendix A.  Applicability to Legacy syslog .................. 11
   A.1.  Method Change ..................................... 12
   A.2.  Octet-Counting .................................... 12
   A.3.  Octet Stuffing .................................... 12
Authors’ Addresses ........................................ 13
1. Introduction

The syslog protocol [RFC5424] is a text-based protocol used to convey event information. Before that standard was produced, syslog messages were being transmitted over UDP. This was described in the INFORMATIONAL document [RFC3164]. While there has been no documented standard for transporting syslog messages over TCP, it is widely used in practice and has proven to be quite interoperable among the implementations, with some minor issues in some configurations. While existing implementations interoperate quite well with each other, there are some differences in protocol handling. This document will describe the most commonly used approach and explain how to interoperate with them in a consistent way.

This specification applies to messages transmitted using the [RFC5424] format. Diagram 1 shows how this relates to the other syslog message transport protocols. 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 TLC transport which is described in [RFC5425]. The collector is shown with the capability to accept all three transports.

A discussion of how this may be applied to legacy syslog as described in [RFC3164] is contained below. Two format options have been observed with legacy syslog being transported over TCP. These are called octet-stuffing and octet-counting and are described in Section 3.3. This specification is written this way, with two format options, in an attempt to ensure that syslog transport receivers can receive and properly interpret messages sent from legacy syslog senders.

To ensure interoperability between syslog senders and receivers, the octet-counting method is REQUIRED and the octet-stuffing method is NOT RECOMMENDED.
Diagram 1. Syslog Layers

There are several advantages to using TCP: flow control, error recovery, reliability, to name a few. 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 to have the sending TCP immediately send all buffered data. [RFC0793]

It is still RECOMMENDED to use the TLS transport [RFC5424] to convey syslog messages. This specification is provided to ensure interoperability for transporting syslog over TCP.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 syslog transported over UDP as described in [RFC3164]. Devices that continue to use that will be described as legacy syslog senders and receivers in this document.

3. Message Transmission

As described in [RFC5424], syslog is simplex in nature. Traditional implementations of syslog over TCP 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. Reliability and flow control are provided by the capabilities of TCP.

3.1. Session

A syslog over TCP session is a TCP connection between a client and a server. The syslog transport sender is the host that sends the original SYN. The syslog transport receiver is the device that receives the original SYN and responds with a SYN+ACK. 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 are fixed once the session is established, and they can not be reversed during the session.
However, there can be multiple sessions between two TCP hosts, and for each session the role of transport sender and transport receiver can be different based upon which device initiates the session.

It is valid (but rare) for no messages to be exchanged during a TCP session.

If an error occurs, the peer detecting the error will gracefully close the TCP session, but has no means to notify its remote peer about the state of the peer syslog application.

### 3.2. Session Initiation

The peer that intends to act as a syslog transport receiver listens to TCP port <TBD>. The peer that intends to act as the transport sender initiates a TCP session to the syslog transport receiver as specified in [RFC0793].

### 3.3. Message Transfer

During the message transfer phase, the syslog transport sender sends a stream of messages to the transport receiver. Either of the peers may initiate session closure at any time as specified in Section 3.5 of [RFC0793]. In practice, this is often seen after a prolonged time of inactivity.

Syslog messages are sent in sequence within a TCP transport stream. One message is encapsulated inside a frame. Syslog transport senders MUST support the octet-counting method and they MAY support the octet-stuffing method. Syslog transport receivers are REQUIRED to support the octet-counting method and are RECOMMENDED to support the octet-stuffing method to promote interoperability with legacy devices that may only use that framing method. Transport senders do not send any notice about the format they use to the transport receiver. However, the format itself enables the transport receiver to detect which framing is used. The syslog transport sender MUST NOT change the format after it has sent the first message. If the format needs to be changed, the TCP session must be concluded and a new session established.

All syslog messages MUST be sent as 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
SYSLOG-FRAME =/ SYSLOG-MSG TRAILER ; Octet-stuffing method

MSG-LEN = NONZERO-DIGIT *DIGIT

SP = %d32
NONZERO-DIGIT = %d49-57
DIGIT = %d48 / NONZERO-DIGIT

TRAILER = LF | APP-DEFINED

LF = %d10
APP-DEFINED = 1*2OCTET

SYSLOG-MSG is defined in the syslog protocol [RFC5424]

3.3.1. Octet-Counting

This mode is somewhat similar to the framing used in [RFC5425]. Here the message length, in octets, is specified as HEADER, followed by SYSLOG-MSG and no trailer.

MSG-LEN is the octet count of the SYSLOG-MSG in the SYSLOG-FRAME. A transport receiver MUST use the message length to delimit a syslog message. There is no upper limit for a message length per se. However, in order to establish a baseline for interoperability, this specification requires that a transport receiver MUST be able to process messages with a length up to and including 2048 octets. Transport receivers SHOULD be able to process messages with lengths up to and including 8192 octets.

A transport receiver MUST assume that octet-counting framing is used if a syslog frame starts with a digit.

3.3.2. Octet-Stuffing

In octet-stuffing mode, there is no header, but a trailer is appended after SYSLOG-MSG. For this specification, this character MUST be the USASCII LF (%d10) character.

A transport receiver MUST accept the USASCII LF character as a TRAILER. It MAY be configurable to accept other characters. A
A transport receiver MUST assume that octet-stuffing framing is used if a syslog frame starts with the USASCII character "<" (%d60). The octet-stuffing method is NOT RECOMMENDED.

3.4. Retaining the Original Message

In both of these methods, a modification is made to the original message. This is a temporary transformation performed by the transport sender. According to Section 5 of [RFC5425], this temporary transformation MUST be reversed by the transport protocol at the transport receiver so that the relay or collector will see an exact copy of the message generated by the originator or relay.

In the octet-counting method, a count and a space character are prepended to the messages. This is very much like the method described in [RFC5426]. The count and space character MUST be removed by the transport receiver after it has validated that the count is correct.

Similarly, the syslog transport receiver MUST discard the TRAILER as it accepts the packet in the octet-stuffing method.

3.5. Session Closure

The SYSLOG session is closed when one of the peers decides to do so. It then initiates a local TCP session closure. It does not notify its remote peer of its intention to close the session, nor does it accept any messages that are still in transit.

4. Security Considerations

Using this specification on an unsecured network is NOT RECOMMENDED. Several syslog security considerations are discussed in [RFC5424]. This section focuses on security considerations specific to the syslog transport over TCP. Some of the security issues raised in this section can be mitigated through the use of TLS as defined in [RFC5425]

4.1. Sender Authentication and Message Forgery

This transport mapping does not provide for strong transport sender authentication. The receiver of the syslog message will not be able to ascertain that the message was indeed sent from the reported sender, or whether the packet was sent from another device. This can
also lead to a case of mistaken identity if an inappropriately configured machine sends syslog messages representing itself as another machine.

This transport mapping does not provide protection against syslog message forgery. An attacker can transmit syslog messages (either from the machine from which the messages are purportedly sent or from any other machine) to a receiver.

In one case, an attacker can hide the true nature of an attack amidst many other messages. As an example, an attacker can start generating forged messages indicating a problem on some machine. This can get the attention of the system administrators, who will spend their time investigating the alleged problem. During this time, the attacker could be able to compromise a different machine or a different process on the same machine.

Additionally, an attacker can generate false syslog messages to give untrue indications of the status of systems. As an example, an attacker can stop a critical process on a machine, which could generate a notification of exit. The attacker can subsequently generate a forged notification that the process had been restarted. The system administrators could accept that misinformation and not verify that the process had indeed not been restarted.

4.2. Message Observation

This transport mapping does not provide confidentiality of the messages in transit. If syslog messages are in clear text, this is how they will be transferred. In most cases, passing clear-text, human-readable messages is a benefit to the administrators. Unfortunately, an attacker could also be able to observe the human-readable contents of syslog messages. The attacker could then use the knowledge gained from these messages to compromise a machine. It is RECOMMENDED that no sensitive information be transmitted via this transport mapping or that transmission of such information be restricted to properly secured networks.

4.3. Replaying

Message forgery and observation can be combined into a replay attack. An attacker could record a set of messages that indicate normal activity of a machine. At a later time, an attacker could remove that machine from the network and replay the syslog messages with new time stamps. The administrators could find nothing unusual in the received messages, and their receipt would falsely indicate normal activity of the machine.
4.4. Message Prioritization and Differentiation

This transport mapping does not mandate prioritization of syslog messages either on the wire or when processed on the receiving host based on their severity. Unless some prioritization is implemented by sender, receiver, and/or network, the security implication of such behavior is that the syslog receiver or network devices could get overwhelmed with low-severity messages and be forced to discard potentially high-severity messages.

4.5. Denial of Service

An attacker could overwhelm a receiver by sending more messages to it than could be handled by the infrastructure or the device itself. Implementers SHOULD attempt to provide features that minimize this threat, such as optionally restricting reception of messages to a set of known source IP addresses.

4.6. Reliability

It should be noted that the syslog transport specified in this document does not use application-layer acknowledgments. TCP uses retransmissions to provide protection against some forms of data loss. However, if the TCP connection is broken for some reason (or closed by the transport receiver), the syslog transport sender cannot always know what messages were successfully delivered to the syslog application at the other end.

5. IANA Considerations

IANA is requested to provide a TCP port for this protocol.

After that port has been assigned, this section will be revised to list that port.

6. Acknowledgments

The authors wish to thank David Harrington and all other people who commented on various versions of this proposal.

7. 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.
Please replace the instances of <TBD> the port number assigned by IANA.

This is version -01 based upon review comments from David Harrington.

8. References

8.1. Normative


8.2. Informative


Appendix A. Applicability to Legacy syslog

This is an informative appendix provided to promote interoperability within the various observed implementations. Even though this specification does not cover legacy syslog messages, the language used here will be consistent with [RFC2119] to be clear in this and to show how the new syslog architecture will interoperate with the legacy implementations.

Syslog over TCP has been around for a number of years. Just like legacy syslog over UDP, several different implementations exist. The older method of octet-stuffing has problems so is NOT RECOMMENDED, but SHOULD be implemented to ensure interoperability with older clients or servers that may only use this method. The newer method
of octet-counting is reliable and, as is consistent with this specification, SHOULD be implemented. When implementers do implement both methods, it is RECOMMENDED that the default method be octet-counting.

A.1. Method Change

It has been observed in legacy implementations that the framing may change on a frame-by-frame basis. This behavior is NOT RECOMMENDED. However, for interoperability, a transport receiver wishing to interoperate with these legacy systems SHOULD be prepared to accept different framing for each frame received.

A.2. Octet-Counting

This framing allows for the transmission of all characters inside SYSLOG-MSG. Along these lines, some transport senders have been seen to use this framing to stack multiple messages within a single TCP frame by using a USASCII LF (%d10) character to separate them. As an example, two messages may be placed within a single frame with a LF character between them. This behavior is NOT RECOMMENDED although it has been observed. Since it cannot be known in advance if the legacy transport sender is going to send multiple messages within single frames, it MUST be assumed that they WILL NOT, and that each frame using this method contains only one syslog message. Implementers MAY attempt to address this by looking for TRAILER characters within each frame to try to separate multiple messages. However, this may lead to more problems than it resolves.

A transport receiver MUST assume that octet-counting framing is used if a syslog frame starts with a digit.

A.3. Octet Stuffing

The problem with octet-stuffing framing comes from the use of [RFC3164] messages. In that, the traditional trailer character is not escaped within SYSLOG-MSG 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 transport receiver. There is no method to avoid this problem with the octet-stuffing framing.

In this legacy implementation, the TRAILER consists of a single character and most often is the USASCII LF (%d10) character. However, other characters have also been seen occasionally, with USACII NUL (%d00) being a prominent example. Some devices also emit a two-character TRAILER, which is usually CR and LF.
Transport senders MUST support the option to use the USASCII LF character. Transport receivers MUST also support this. Transport senders and receivers MAY also support other characters.

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