Network Working Group
Request for Comments: 5426
Category: Standards Track
A. Okmianski
Cisco Systems, Inc.
March 2009

Transmission of Syslog Messages over UDP

Status of This Memo

Copyright Notice

Abstract

This document describes the transport for syslog messages over UDP/IPv4 or UDP/IPv6. The syslog protocol layered architecture provides for support of any number of transport mappings. However, for interoperability purposes, syslog protocol implementers are required to support this transport mapping.

Table of Contents

1. Introduction

Informational RFC 3164 [8] describes the syslog protocol as it was observed in existing implementations. It describes both the format of syslog messages and a UDP [1] transport. Subsequently, a Standards-Track syslog protocol has been defined in RFC 5424 [2].

RFC 5424 specifies a layered architecture that provides for support of any number of transport layer mappings for transmitting syslog messages. This document describes the UDP transport mapping for the syslog protocol.

The transport described in this document can be used for transmitting syslog messages over both IPv4 [3] and IPv6 [4].

Network administrators and architects should be aware of the significant reliability and security issues of this transport, which stem from the use of UDP. They are documented in this specification. However, this transport is lightweight and is built upon the existing popular use of UDP for syslog.

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 [5].

3. Transport Protocol

3.1. One Message Per Datagram

Each syslog UDP datagram MUST contain only one syslog message, which MAY be complete or truncated. The message MUST be formatted and truncated according to RFC 5424 [2]. Additional data MUST NOT be present in the datagram payload.

3.2. Message Size

This transport mapping supports transmission of syslog messages up to 65535 octets minus the UDP header length. This limit stems from the maximum supported UDP size of 65535 octets specified in RFC 768 [1]. For IPv4, the maximum payload size is 65535 octets minus the UDP header and minus the IP header because IPv4 has a 16-bit length field that also includes the header length.

IPv4 syslog receivers MUST be able to receive datagrams with message sizes up to and including 480 octets. IPv6 syslog receivers MUST be able to receive datagrams with message sizes up to and including 1180 octets. All syslog receivers SHOULD be able to receive datagrams with message sizes of up to and including 2048 octets. The ability to receive larger messages is encouraged.

The above restrictions and recommendations establish a baseline for interoperability. The minimum required message size support was determined based on the minimum MTU size that Internet hosts are required to support: 576 octets for IPv4 [3] and 1280 octets for IPv6 [4]. Datagrams that conform to these limits have the greatest chance of being delivered because they do not require fragmentation.

It is RECOMMENDED that syslog senders restrict message sizes such that IP datagrams do not exceed the smallest MTU of the network in use. This avoids datagram fragmentation and possible issues surrounding fragmentation such as incorrect MTU discovery.

Fragmentation can be undesirable because it increases the risk of the message being lost due to loss of just one datagram fragment. Syslog has no acknowledgement facility, and therefore there is no effective way to handle retransmission. This makes it impossible for syslog to utilize packetization layer path MTU discovery [9]. When network MTU is not known in advance, the safest assumption is to restrict messages to 480 octets for IPv4 and 1180 octets for IPv6.

3.3. Source and Target Ports

Syslog receivers MUST support accepting syslog datagrams on the well-known UDP port 514, but MAY be configurable to listen on a different port. Syslog senders MUST support sending syslog message datagrams to the UDP port 514, but MAY be configurable to send messages to a different port. Syslog senders MAY use any source UDP port for transmitting messages.

3.4. Source IP Address

The source IP address of the UDP datagrams SHOULD NOT be interpreted as the identifier for the host that originated the syslog message. The entity sending the syslog message could be merely a relay. The syslog message itself contains the identifier of the originator of the message.

3.5. UDP/IP Structure

Each UDP/IP datagram sent by the transport layer MUST completely adhere to the structure specified in the UDP RFC 768 [1] and either the IPv4 RFC 791 [3] or IPv6 RFC 2460 [4], depending on which protocol is used.

3.6. UDP Checksums

Syslog senders MUST NOT disable UDP checksums. IPv4 syslog senders SHOULD use UDP checksums when sending messages. Note that RFC 2460 [4] mandates the use of UDP checksums when sending UDP datagrams over IPv6.

Syslog receivers MUST NOT disable UDP checksum checks. IPv4 syslog receivers SHOULD check UDP checksums and SHOULD accept a syslog message with a zero checksum. Note that RFC 2460 [4] mandates the use of checksums for UDP over IPv6.

4. Reliability Considerations

The UDP is an unreliable, low-overhead protocol. This section discusses reliability issues inherent in UDP that implementers and users should be aware of.

4.1. Lost Datagrams

This transport mapping does not provide any mechanism to detect and correct loss of datagrams. Datagrams can be lost in transit due to congestion, corruption, or any other intermittent network problem. IP fragmentation exacerbates this problem because loss of a single fragment will result in the entire message being discarded.

4.2. Message Corruption

The UDP/IP datagrams can get corrupted in transit due to software, hardware, or network errors. This transport mapping specifies use of UDP checksums to enable corruption detection in addition to checksums used in IP and Layer 2 protocols. However, checksums do not guarantee corruption detection, and this transport mapping does not provide for message acknowledgement or retransmission mechanism.

4.3. Congestion Control

Because syslog can generate unlimited amounts of data, transferring this data over UDP is generally problematic, because UDP lacks congestion control mechanisms. Congestion control mechanisms that respond to congestion by reducing traffic rates and establish a degree of fairness between flows that share the same path are vital to the stable operation of the Internet [6]. This is why the syslog TLS transport [7] is REQUIRED to implement and RECOMMENDED for general use.

The only environments where the syslog UDP transport MAY be used as an alternative to the TLS transport are managed networks, where the network path has been explicitly provisioned for UDP syslog traffic through traffic engineering mechanisms, such as rate limiting or capacity reservations. In all other environments, the TLS transport [7] SHOULD be used.

4.4. Sequenced Delivery

The IP transport used by the UDP does not guarantee that the sequence of datagram delivery will match the order in which the datagrams were sent. The time stamp contained within each syslog message can serve as a rough guide in establishing sequence order. However, it will not help in cases where multiple messages were generated during the same time slot, the sender could not generate a time stamp, or messages originated from different hosts whose clocks were not synchronized. The order of syslog message arrival via this transport SHOULD NOT be used as an authoritative guide in establishing an absolute or relative sequence of events on the syslog sender hosts.

5. Security Considerations

Using this specification on an unsecured network is NOT RECOMMENDED. Several syslog security considerations are discussed in RFC 5424 [2]. This section focuses on security considerations specific to the syslog transport over UDP. Some of the security issues raised in this section can be mitigated through the use of IPsec as defined in RFC 4301 [10].

5.1. Sender Authentication and Message Forgery

This transport mapping does not provide for strong 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 to a receiver 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.

5.2. 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.

5.3. 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.

5.4. Unreliable Delivery

As was previously discussed in Section 4, Reliability Considerations, the UDP transport is not reliable, and packets containing syslog message datagrams can be lost in transit without any notice. There can be security consequences to the loss of one or more syslog messages. Administrators could be unaware of a developing and potentially serious problem. Messages could also be intercepted and discarded by an attacker as a way to hide unauthorized activities.

5.5. 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.

5.6. 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.

6. IANA Considerations

This transport uses UDP port 514 for syslog, as recorded in the IANA port-numbers registry.

7. Acknowledgements

The author gratefully acknowledges the contributions of: Chris Lonvick, Rainer Gerhards, David Harrington, Andrew Ross, Albert Mietus, Bernie Volz, Mickael Graham, Greg Morris, Alexandra Fedorova, Devin Kowatch, Richard Graveman, and all others who have commented on the various versions of this proposal.

8. References

8.1. Normative References

[1] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

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

[3] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[4] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[6] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[7] Miao, F. and Y. Ma, "TLS Transport Mapping for Syslog", RFC 5425, March 2009.

8.2. Informative References

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

[9] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[10] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Author's Address

Anton Okmianski
Cisco Systems, Inc.
595 Burrard St., Suite 2123
Vancouver, BC V7X 1J1
Canada

Phone: +1-978-936-1612
EMail: aokmians@cisco.com