Internet Engineering Task Force (IETF)
Request for Comments: 6012
Category: Standards Track
ISSN: 2070-1721
October 2010

J. Salowey
Cisco Systems, Inc.
T. Petch
Engineering Networks Ltd
R. Gerhards
Adiscon GmbH
H. Feng
Huaweisymantec Technologies

Datagram Transport Layer Security (DTLS) Transport Mapping for Syslog

Abstract

This document describes the transport of syslog messages over the Datagram Transport Layer Security (DTLS) protocol. It provides a secure transport for syslog messages in cases where a connectionless transport is desired.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6012.

Copyright Notice

Table of Contents

1. Introduction

The syslog protocol [RFC5424] is designed to run over different transports for different environments. This document defines the transport of syslog messages over the Datagram Transport Layer Security (DTLS) protocol [RFC4347].

The Datagram Transport Layer Security (DTLS) protocol [RFC4347] is designed to meet the requirements of applications that need secure datagram transport. DTLS has been mapped onto different transports, including UDP [RFC0768] and the Datagram Congestion Control Protocol (DCCP) [RFC4340]. This memo defines both options, namely syslog over DTLS over UDP, and syslog over DTLS over DCCP.

2. Terminology

The following definitions from [RFC5424] are used in this document:

- An "originator" generates syslog content to be carried in a message.
- A "collector" gathers syslog content for further analysis.
- A "relay" forwards messages, accepting messages from originators or other relays, and sending them to collectors or other relays.
- A "transport sender" passes syslog messages to a specific transport protocol.
- A "transport receiver" takes syslog messages from a specific transport protocol.

This document adds the following definitions:

- A "DTLS client" is an application that can initiate a DTLS Client Hello to a server.
- A "DTLS server" is an application that can receive a DTLS Client Hello from a client and reply with a Server Hello.

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 [RFC2119].

3. Security Requirements for Syslog

The security requirements for the transport of syslog messages are discussed in Section 2 of [RFC5425]. These also apply to this specification.

The following secondary threat is also considered in this document:

- o Denial of service is discussed in [RFC5424], which states that an attacker may send more messages to a transport receiver than the transport receiver could handle. When using a secure transport protocol handshake, an attacker may use a spoofed IP source to engage the server in a cryptographic handshake to deliberately consume the server's resources.

4. Using DTLS to Secure Syslog

DTLS can be used as a secure transport to counter all the primary threats to syslog described in [RFC5425]:

- o Confidentiality to counter disclosure of the message contents.
- o Integrity checking to counter modifications to a message on a hop-by-hop basis.
- o Server or mutual authentication to counter masquerade.

In addition, DTLS also provides:

- o A cookie exchange mechanism during handshake to counter Denial of Service attacks.
- o A sequence number in the header to counter replay attacks.

Note: This secure transport (i.e., DTLS) only secures syslog transport in a hop-by-hop manner, and is not concerned with the contents of syslog messages. In particular, the authenticated identity of the transport sender (e.g., subject name in the certificate) is not necessarily related to the HOSTNAME field of the syslog message. When authentication of syslog message origin is required, [RFC5848] can be used.

5. Protocol Elements

5.1. Transport

DTLS can run over multiple transports. Implementations of this specification MUST support DTLS over UDP and SHOULD support DTLS over DCCP [RFC5238]. Transports such as UDP or DCCP do not provide session multiplexing and session demultiplexing. In such cases, the application implementer provides this functionality by mapping a unique combination of the remote address, remote port number, local address, and local port number to a session.

Each syslog message is delivered by the DTLS record protocol, which assigns a sequence number to each DTLS record. Although the DTLS implementer may adopt a queue mechanism to resolve reordering, it may not assure that all the messages are delivered in order when mapping on the UDP transport.

When DTLS runs over an unreliable transport, such as UDP, reliability is not provided. With DTLS, an originator or relay may not realize that a collector has gone down or lost its DTLS connection state, so messages may be lost.

Syslog over DTLS over TCP MUST NOT be used. If a secure transport is required with TCP, then the appropriate security mechanism is syslog over Transport Layer Security (TLS) as described in [RFC5425].

5.2. Port and Service Code Assignment

A syslog transport sender is always a DTLS client, and a transport receiver is always a DTLS server.

The UDP and DCCP port 6514 has been allocated as the default port for syslog over DTLS as defined in this document. The service code SYLG (1398361159) has been assigned to syslog.

5.3. Initiation

The transport sender initiates a DTLS connection by sending a DTLS Client Hello to the transport receiver. Implementations MUST support the denial of service countermeasures defined by DTLS. When these countermeasures are used, the transport receiver responds with a DTLS Hello Verify Request containing a cookie. The transport sender responds with a DTLS Client Hello containing the received cookie, which initiates the DTLS handshake. The transport sender MUST NOT send any syslog messages before the DTLS handshake has successfully completed.

Implementations MUST support DTLS 1.0 [RFC4347] and MUST support the mandatory to implement cipher suite, which is TLS_RSA_WITH_AES_128_CBC_SHA as specified in [RFC5246]. If additional cipher suites are supported, then implementations MUST NOT negotiate a cipher suite that employs NULL integrity or authentication algorithms.

Where privacy is REQUIRED, then implementations must either negotiate a cipher suite that employs a non-NULL encryption algorithm or else achieve privacy by other means, such as a physically secured network.

However, as [RFC5424], Section 8, points out, "In most cases, passing clear-text messages is a benefit to the operations staff if they are sniffing the packets from the wire", and so where privacy is not a requirement, then it is advantageous to use a NULL encryption algorithm.

5.3.1. Certificate-Based Authentication

The mandatory to implement cipher suites for DTLS use certificates [RFC5280] to authenticate peers. Both the syslog transport sender (DTLS client) and the syslog transport receiver (DTLS server) MUST implement certificate-based authentication. This consists of validating the certificate and verifying that the peer has the corresponding private key. The latter part is performed by DTLS. To ensure interoperability between clients and servers, the methods for certificate validation defined in Sections 4.2.1 and 4.2.2 of [RFC5425] SHALL be implemented.

Both transport receiver and transport sender implementations MUST provide means to generate a key pair and self-signed certificate in case a key pair and certificate are not available through another mechanism.

The transport receiver and transport sender SHOULD provide mechanisms to record the certificate or certificate fingerprint used by the remote endpoint for the purpose of correlating an identity with the sent or received data.

5.4. Sending Data

All syslog messages MUST be sent as DTLS "application data". It is possible that multiple syslog messages be contained in one DTLS record, or that a syslog message be transferred in multiple DTLS records. The application data is defined with the following ABNF [RFC5234] expression:

```
APPLICATION-DATA = 1*SYSLOG-FRAME

SYSLOG-FRAME = MSG-LEN SP SYSLOG-MSG

MSG-LEN = NONZERO-DIGIT *DIGIT

SP = %d32

NONZERO-DIGIT = %d49-57

DIGIT = %d48 / NONZERO-DIGIT
```

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

5.4.1. Message Size

The message length 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. As stated in [RFC4347], a DTLS record MUST NOT span multiple datagrams. When mapping onto different transports, DTLS has different record size limitations. For UDP, see Section 3.2 of [RFC5426]. For DCCP, the application implementer SHOULD determine the maximum record size allowed by the DTLS protocol running over DCCP, as specified in [RFC4340]. The message size SHOULD NOT exceed the DTLS maximum record size limitation of 2^14 bytes. To be consistent with [RFC5425], in establishing 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.

See Appendix A.2 of [RFC5424] for implementation guidance on message length, including fragmentation.

5.5. Closure

A transport sender MUST close the associated DTLS connection if the connection is not expected to deliver any syslog messages later. It MUST send a DTLS close_notify alert before closing the connection. A transport sender (DTLS client) MAY choose to not wait for the transport receiver's close_notify alert and simply close the DTLS connection. Once the transport receiver gets a close_notify from the transport sender, it MUST reply with a close_notify.

When no data is received from a DTLS connection for a long time (where the application decides what "long" means), a transport receiver MAY close the connection. The transport receiver (DTLS server) MUST attempt to initiate an exchange of close_notify alerts with the transport sender before closing the connection. Transport receivers that are unprepared to receive any more data MAY close the connection after sending the close_notify alert.

Although closure alerts are a component of TLS and so of DTLS, they, like all alerts, are not retransmitted by DTLS and so may be lost over an unreliable network.

6. 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 establishing a degree of fairness between flows that share the same path are vital to the stable operation of the Internet (see [RFC2914] and [RFC5405]).

DCCP has congestion control. If DCCP is available, syslog over DTLS over DCCP is RECOMMENDED in preference to syslog over DTLS over UDP. Implementations of syslog over DTLS over DCCP MUST support Congestion Control Identifier (CCID) 3 and SHOULD support CCID 2 to ensure interoperability.

The congestion control considerations from Section 4.3 of [RFC5426] also apply to syslog over DTLS over UDP.

7. Security Policies

Syslog transport over DTLS has been designed to minimize the security and operational differences for environments where both syslog over TLS [RFC5425] and syslog over DTLS are supported. The security policies for syslog over DTLS are the same as those described in [RFC5425], and all the normative requirements of Section 5 of [RFC5425] apply.

8. IANA Considerations

IANA has assigned a registered UDP and DCCP port number for syslog over DTLS. The values are the same as for syslog over TLS. That is, the registry has been updated as follows:

```
syslog-tls    6514/udp    syslog over DTLS [RFC6012]
syslog-tls    6514/dccp   syslog over DTLS [RFC6012]
```

IANA has assigned the service code SYLG to syslog for use with DCCP. The allocation in the Service Code subregistry of the Datagram Congestion Control Protocol (DCCP) Parameters registry is as follows:

```
1398361159    SYLG    Syslog Protocol    [RFC6012]
```

9. Security Considerations

The security considerations in [RFC4347], [RFC5246], [RFC5425], and [RFC5280] apply to this document.

9.1. DTLS Renegotiation

TLS and DTLS renegotiation may be vulnerable to attacks described in [RFC5746]. Although RFC 5746 provides a fix for some of the issues, renegotiation can still cause problems for applications since connection security parameters can change without the application knowing it. Therefore it is RECOMMENDED that renegotiation be disabled for syslog over DTLS. If renegotiation is allowed, then the specification in RFC 5746 MUST be followed, and the implementation MUST make sure that the connection still has adequate security and that any identities extracted from client and server certificates do not change during renegotiation.

9.2. Message Loss

The transports described in this document are unreliable. It is possible for messages to be lost or removed by an attacker without the knowledge of the receiver. [RFC5424] notes that implementers who wish a lossless stream should be using tls/tcp as their transport. In addition, the use of signed syslog messages [RFC5848] can also provide an indication of message loss.

9.3. Private Key Generation

Transport receiver and transport sender implementations often generate their own key pairs. An inadequate random number generator (RNG) or an inadequate pseudo-random number generator (PRNG) to generate these keys can result in little or no security. See [RFC4086] for random number generation guidance.

9.4. Trust Anchor Installation and Storage

Trust anchor installation and storage is critical. Transmission of a trust anchor, especially self-signed certificates used as trust anchors, from transport receiver to transport sender for installation requires one or more out-of-band steps. Care must be taken to ensure the installed trust anchor is in fact the correct trust anchor. The fingerprint mechanism mentioned in Section 5.3.1 can be used by the transport sender to ensure the transport receiver's self-signed certificate is properly installed. Trust anchor information must be securely stored. Changes to trust anchor information can cause acceptance of certificates that should be rejected.

10. Acknowledgements

The authors would like to thank Wes Hardaker for his review of this proposal and for contributing his valuable suggestions on the use of DTLS. Thanks also to Pasi Eronen, David Harrington, Chris Lonvick, Eliot Lear, Anton Okmyanskiy, Juergen Schoenwaelder, Richard Graveman, the members of the syslog working group, and the members of the IESG for their review, comments, and suggestions.

11. References

11.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

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

[RFC5238] Phelan, T., "Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP)", RFC 5238, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 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.

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

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, <"Transport Layer Security (TLS) Renegotiation Indication Extension", RFC 5746, February 2010.

11.2. Informative References

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

[RFC5848] Kelsey, J., Callas, J., and A. Clemm, "Signed Syslog Messages", RFC 5848, May 2010.

Authors' Addresses

Joseph Salowey
Cisco Systems, Inc.
2901 3rd Ave.
Seattle, WA 98121
USA

EMail: jsalowey@cisco.com

Tom Petch
Engineering Networks Ltd
18 Parkwood Close
Lymm, Cheshire WA13 0NQ
UK

EMail: tomSecurity@network-engineer.co.uk

Rainer Gerhards
Adiscon GmbH
Mozartstrasse 21
Grossrinderfeld, BW 97950
Germany

EMail: rgerhards@adiscon.com

Hongyan Feng
Huaweisymantec Technologies
20245 Stevens Creek Blvd.
Cupertino, CA 95014

EMail: fhyfeng@gmail.com