DTLS transport mapping for SYSLOG
draft-petch-gerhards-syslog-transport-dtls-02.txt

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 December 7, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document describes the transport of syslog messages over DTLS (Datagram Transport Level Security). It provides a secure transport
for syslog messages in cases where a connection-less transport is desired.

Table of Contents

1. Introduction .................................................. 3
   1.1. Conventions Used in This Document ........................ 3
   1.2. Terminology ................................................. 3
   1.3. Security requirements for syslog .......................... 3
   1.4. TLS ....................................................... 4
   1.5. DTLS ..................................................... 5
   1.6. Client and server ......................................... 6
2. DTLS usage .................................................... 7
   2.1. Authentication ............................................ 7
   2.2. Port number ............................................... 7
   2.3. Invoking DTLS ............................................. 8
   2.4. DTLS connection termination .............................. 8
   2.5. Delineation of datagrams ................................. 9
   2.6. Choice of ciphersuite ................................... 9
   2.7. Fragmentation ............................................ 10
   2.8. TLS extensions .......................................... 10
3. Security Considerations ....................................... 11
4. Authors ........................................................ 11
5. IANA Considerations ........................................... 12
6. Acknowledgments ............................................... 12
7. References .................................................... 12
   7.1. Normative ................................................ 12
   7.2. Informative .............................................. 13
Authors’ Addresses .............................................. 13
1. Introduction

The syslog protocol allows a host to send event notification messages across the Internet to another host. This memo describes the use of DTLS [I-D.ietf-tls-rfc4347-bis] to secure that syslog traffic, using UDP as the transport protocol.

The memo was first published in 2006 and was republished in March 2009 in the light of renewed interest in the topic. That update included only changes necessary to get through the submission process. This update brings the text more in line with that of current, related memos (such as [RFC5424] and [RFC5539]) but more work is needed in this area, particularly with regard to host identity checking.

DTLS is TLS1.2 [RFC5246] modified as little as possible; understanding DTLS requires an understanding of TLS. This introduction provides an overview of TLS followed by an overview of the modifications that DTLS makes, as well as defining syslog terminology and syslog security requirements. The rest of the memo defines the use of TLS/DTLS features to secure syslog traffic.

1.1. 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 RFC2119 [RFC2119].

1.2. Terminology

The following definitions from [RFC5424] are used in this memo

- 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.

A single application can have multiple roles at the same time.

1.3. Security requirements for syslog

As a result of discussions on the syslog-sec mailing list, three primary threats have been identified for syslog:
Modification of information; the contents of a message are altered in transit between a transport sender and a transport receiver, either maliciously or accidentally.

Masquerade; messages are sent by, or sent to, the 'wrong' party masquerading as the intended party. Masquerade of a transport sender of syslog messages is seen as of greater concern than masquerade of a transport receiver ie the latter may not be considered a threat in some environments.

Disclosure; the contents of messages are disclosed when they should not be, for example, revealing the security credentials of a user. The extent to which this is a threat depends on the content of the syslog message and this action may not be considered a threat in some environments.

Traffic Analysis and Denial of Service are not considered threats to syslog.

1.4. TLS

DTLS is TLS modified as little as possible to run over an unreliable transport.

TLS1.2 [RFC5246] can provide a secure transport connection between two endpoints and can provide protection against all the threats to syslog identified above. It runs as a shim, between the application layer and a reliable transport (normally TCP), and sets up a tunnel between the endpoints. Using TCP as that transport brings more than reliability – eg error detection and recovery, flow control, connections – which may not be suitable for the application; DTLS offers the option of TLS over UDP, adding reliability to UDP but without all the additional features that TCP brings. As such, it offers an attractive security option for UDP-based applications.

TLS negotiates options for compression, key exchange, authentication and encryption. Compression is negotiated per se and, while likely to be a desirable feature for the character-oriented syslog, is not seen as an aspect of security and so will not be considered further here. The other three options come as a set, one of a number of predefined ciphersuites, and it is the ciphersuite, not an individual option, that is negotiated. The currently defined ciphersuites can be found on the IANA website [IANA].

Thus the ciphersuite known as

TLS_RSA_WITH_AES_128_CBC_SHA

uses AES_128 with CBC for encryption, SHA when a hash is required and a PKI certificate with an RSA public key valid for encryption as the
basis for end system authentication. This is the default ciphersuite, the one that an application using TLS is REQUIRED to implement, for inter-operability, unless the usage of TLS by the application is specified otherwise in an RFC.

Different ciphersuites provide different levels of security with respect to each of key exchange, message authentication and encryption, ranging from none through weak to strong. Security is a dynamic field, with techniques being reclassified from strong to weak, eg as a result of advances in mathematics or of an increase in the computing power available to a potential attacker. At the same time, new cryptographic techniques appear, and are in turn incorporated into TLS (and other) ciphersuites.

The user of TLS, whether for syslog or any other application, MUST verify that the ciphersuite in use provides adequate security for their particular environment at the time that TLS is used. Any syslog application invoking TLS SHOULD verify that the ciphersuite in use meets the minimum standards of the application and MAY provide customisation to specify a minimum acceptable one.

Disclosure is identified as a threat to syslog and countering this implies the use of encryption. If, in a particular environment, disclosure is not regarded as a threat, as when the network is physically secure, then the use of a ciphersuite with NULL encryption would be appropriate; this may be a consideration where the syslog client is a device with limited processing capability.

1.5. DTLS

DTLS is TLS, modified, as little as possible, to run over an unreliable transport (eg UDP, DCCP). TLS decryption requires messages to arrive in sequence so DTLS adds a sequence number to the record header in order to detect reordering. The TLS handshake requires reliable delivery so DTLS specifies timeouts to detect packet loss.

DTLS records are required to fit into a single datagram. This represents no change for a UDP application - eg syslog - but is an issue for the TLS handshake protocol, where, as RFC4347bis [I-D.ietf-tls-rfc4347-bis] points out, messages are, in practice, several kilobytes (cf, eg, a certificate chain). DTLS adds fragmentation to the handshake protocol. Multiple application records may appear in a single DTLS datagram but must not span DTLS datagrams.

Finally, DTLS adds to the transport the concept of a "connection", starting with the TLS handshake, ending with TLS closure alerts.
1.6. Client and server

TLS evolved in the context of HTTP (with the protocol identifier https: [RFC2818]), an environment of powerful servers and relatively powerful workstations. As such, the cost and use of the security algorithms - eg public key encryption, certificate chains - is of limited impact. The security is asymmetric, with server authentication to client as mandatory (assuming that the ciphersuite specifies authentication) while client authentication is optional. This also fits well with HTTP where the client, commonly a human user of a workstation, wants to confirm that they are talking to the server that they think they are and so receive a server certificate which can then be verified. [RFC2818] specifies that if the HTTP client cannot validate the certificate it is offered, then it should pass the decision to a user, or, where there is no user, then it should terminate the connection.

This model does not carry across well to syslog. A syslog device may be a powerful file or Web server, but may, on the other hand, be a low powered, unattended entry level network device, such as remotely located CPE (Customer Premises Equipment), ill-suited to verifying certificate chains and unlikely to have a human user to pass a decision on to.

TLS is now used on less well-equipped devices, such as mobiles; extensions to TLS have been defined which mitigate the impact on the TLS client (eg by using URLs of certificates rather than the certificates themselves). First published separately, the principle of, but not the detail of, these extensions has now been incorporated in the base specification TLS1.2 [RFC5246]: the detailed specification is in RFC4366bis [I-D.ietf-tls-rfc4366-bis]. The asymmetric approach to authentication remains, with server authentication mandatory, client authentication optional.

Most applications which now run over TLS were previously running over TCP and as such already had an application level dialog. In order to invoke TLS, these applications could then change their start command (eg STARTTLS) or, having established a TCP connection, invoke TLS (eg with an AUTH command) and so make the connection secure.

By contrast, syslog uses simplex UDP, a connectionless transport, with syslog messages arriving as and when, independently. DTLS adds the concept of a "connection". The decision to create a connection can be implicit, eg when the first message is sent; syslog client or server may need to decide when to take down the connection.

syslog defines a client and server, the client being a transport sender, the server being a transport receiver. DTLS will also have a
client and server as does UDP. There is a choice as to which syslog entity is the client, and which the server, both for the DTLS and UDP protocols.

The use of syslog over DTLS must address the following issues:

- authentication
- connection set up
- connection termination
- choice of ciphersuite
- choice of TLS extensions
- delineation of syslog datagrams
- invoking DTLS
- fragmentation

2. DTLS usage

2.1. Authentication

DTLS is an asymmetric protocol. Server authentication is required (when authentication is part of the ciphersuite), client authentication is optional. For syslog, the greater threat is perceived as an unauthenticated syslog client generating spurious messages, as opposed to an unauthenticated syslog server receiving them ie it is the authentication of the syslog client that is the more important. Hence it is RECOMMENDED that the syslog server should be the DTLS client. If Masquerade of the syslog server is considered a threat in a particular environment, then the syslog client SHOULD request authentication of the syslog server/DTLS client.

A consequence of this mapping, of DTLS client to syslog server, is that where certificates are used for server authentication, then the syslog server is the one that has to verify the syslog client’s certificates (something that it is likely to have the greater resources to do). The syslog client must have a certificate; the syslog server certificate remains optional.

2.2. Port number

syslog over UDP [RFC5426] has been allocated port 514 while syslog-conn, which uses BEEP [RFC3080], has been allocated port 601 over TCP [RFC3195]. IANA has also allocated port 601 over UDP on the basis of [RFC3195] although that RFC makes no mention of such a usage. syslog over TLS [RFC5425] has been allocated port 6514 over TCP. IANA has reserved port 6514 over UDP.
Earlier versions of this memo proposed the use of port 601 for syslog over DTLS in preference to asking IANA to assign a completely new port. (Ports are a scarce resource, especially those with values of 1024 or less, and their use should be conserved). Now that syslog over TLS has been standardised with the use of port 6514, a request for IANA to allocate 6514 over UDP would appear the best option.

2.3. Invoking DTLS

The DTLS "connection" SHOULD be initiated by syslog client by sending the plain text application level command

AUTH TLS SERVER

when it wishes to be the DTLS server or

AUTH TLS CLIENT

when it wishes to be the DTLS client. The former is expected to be the usual case. (The use of DTLS, as opposed to TLS, is implicit in the use of UDP transport).

The syslog server MUST respond

OK

if it accepts its proposed role or

ERROR

if does not.

This is followed by TLS negotiation with syslog server/DTLS client sending DTLS Client Hello and, if the negotiation is successful, by syslog messages.

2.4. DTLS connection termination

DTLS includes a mechanism for graceful shutdown - TLS1.2 [RFC5246] s.7.2.1. Closure alerts - and these SHOULD be used to terminate a DTLS connection. As specified there, either DTLS client or DTLS server may initiate a closure when it SHOULD send a close_notify alert. Any data received after a closure alert is ignored. The other party MUST send a close_notify alert of its own and close down the connection immediately, discarding any pending writes. The initiator of the close need not wait for the responding close_notify alert before closing the read side of the connection.
Closure should be initiated when the syslog application determines that no more messages will be sent or received, or that none have been for a period of time. A suggested value for the period of time is one hour. The choice of a value should balance the resources needed to re-create the connection using DTLS against the resources used by an idle connection and the increased risk of a breach of security by keeping a DTLS connection in place longer than necessary.

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

2.5. Delineation of datagrams

When syslog runs upon UDP, the UDP datagram frames the syslog message. Over DTLS, syslog messages MUST be sent as DTLS application data. DTLS may place multiple DTLS records in a single datagram, each encoded consecutively. The boundaries are then determined by DTLS record framing.

2.6. Choice of ciphersuite

The Ciphersuite

TLS_RSA_WITH_AES_128_CBC_SHA

is REQUIRED. Where Disclosure is not a threat to syslog and so encryption is not necessary, and may be undesirable because of the limited processing capability of the syslog client, then

TLS_RSA_WITH_NULL_SHA

is RECOMMENDED. Where Pre-Shared Keys are a sufficiently strong security credential, in contrast to the more powerful X.509 Certificates, then

TLS_PSK_WITH_AES_128_CBC_SHA

is RECOMMENDED.

DTLS is defined to use the same ciphersuites as were then current for TLS but excluding those using the stream cipher RC4. New ciphersuites must specify whether or not they are suitable for DTLS and so suitable for use here.
2.7. Fragmentation

syslog messages have no upper limit on size per se and in some environments may be up to $2^{16}$. TLS v1.2 limits messages to $2^{14}$; TLS extensions allow this to be reduced to $2^9$ although, as TLS1.2 [RFC5246] points out, this may be less than needs to be sent at once and so cause the "connection" to be terminated. syslog applications SHOULD ensure that syslog messages fit within the limits of DTLS.

2.8. TLS extensions

TLS Extensions [I-D.ietf-tls-rfc4366-bis] provides specific extensions suitable for a constrained environment. The general extension mechanism ensures backward compatibility for devices that either do not support extensions in general or do not support a particular extension. TLS Extensions identifies wireless as a constrained environment but the constraints, such as limited processing capability or limited storage, apply elsewhere, such as with network devices supporting syslog; TLS Extensions also assumes that it is the TLS client that is constrained, not the TLS server, whereas with network devices, it is the syslog client, here proposed to be TLS server, that is more likely to be constrained. Hence the extensions are of limited relevance.

These extensions are not mentioned in RFC4347bis [I-D.ietf-tls-rfc4347-bis] but, on the basis that DTLS is TLS modified as little as possible, are assumed to be part of DTLS.

The defined extensions, beside the maximum fragment length which has already been discussed, are

- server name indication
- truncated HMAC
- client certificate url
- trusted CA
- certificate status request.

Since syslog over DTLS runs over a port specific to syslog, the server name indication, which helps the DTLS server in identifying the appropriate application, is not required.

Truncated HMAC reduces the HMAC to 80 bit, saving bandwidth, and is not seen as relevant to syslog.

Client certificate URL allows the client, but not the server, to use a URL instead of a certificate per se. Likewise trusted CA allows the client to indicate which CAs it may use so that the server can select a suitable certificate to send. Certificate status request
proposes the use of OCSP instead of CRL. Each of these three cases is designed to help the DTLS client, the syslog server, and so are of limited value since it is expected that it will be the DTLS server, syslog client, that is constrained.

3. Security Considerations

The whole of this memo is about security. As pointed out above, DTLS can provide security against the threats to syslog identified above but whether or not that is achieved depends on the ciphersuite in use. Users MUST ensure that the ciphersuite in use meets their needs for security, in terms of the strength of the algorithms used, whether or not encryption is used and the nature of credentials used to authenticate the parties involved.

The security considerations described in TLS1.2 [RFC5246] apply here.

In addition, the authentication of syslog client and/or server is a significant element in meeting the threat of masquerade. This checking of host identity has been specified in several memos, and, like cryptography in general, is a changing field. The IETF may produce a common standard for this to which this memo (and others) may refer, as opposed to the current practice of each memo producing its own variant. The precise nature of the checks performed are likely to remain a matter of policy for the environment in which syslog over DTLS is used. In this regard, [RFC5539] specifies checks that MUST be performed on a certificate. This MAY include the use of fingerprints as described in [RFC5425].

4. Authors

The authors of this draft are:
5. IANA Considerations

There are no IANA considerations.

6. Acknowledgments

This document was written using the xml2rfc tool described in [RFC2629].

7. References

7.1. Normative


[I-D.ietf-tls-rfc4347-bis]  
Rescorla, E. and N. Modadugu, "Datagram Transport Layer


7.2. Informative


Authors’ Addresses

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