Network Working Group
Request for Comments: 5425
Category: Standards Track

F. Miao, Ed.
Y. Ma, Ed.
Huawei Technologies
J. Salowey, Ed.
Cisco Systems, Inc.

March 2009

Transport Layer Security (TLS) Transport Mapping for Syslog

Status of This Memo

Copyright Notice

Abstract

This document describes the use of Transport Layer Security (TLS) to provide a secure connection for the transport of syslog messages. This document describes the security threats to syslog and how TLS can be used to counter such threats.

Table of Contents

1. Introduction

This document describes the use of Transport Layer Security (TLS [RFC5246]) to provide a secure connection for the transport of syslog [RFC5424] messages. This document describes the security threats to syslog and how TLS can be used to counter such threats.

1.1. Terminology

The following definitions 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.

- A "TLS client" is an application that can initiate a TLS connection by sending a Client Hello to a server.

- A "TLS server" is an application that can receive a 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 RFC 2119 [RFC2119].

2. Security Requirements for Syslog

Syslog messages may transit several hops to arrive at the intended collector. Some intermediary networks may not be trusted by the originator, relay, or receiver because the network is in a different security domain or at a different security level from the originator, relay, or collector. Another security concern is that the originator, relay, or receiver itself is in an insecure network.

There are several threats to be addressed for syslog security. The primary threats are:

- Masquerade. An unauthorized transport sender may send messages to a legitimate transport receiver, or an unauthorized transport receiver may try to deceive a legitimate transport sender into sending syslog messages to it.

- Modification. An attacker between the transport sender and the transport receiver may modify an in-transit syslog message and then forward the message to the transport receiver. Such modification may make the transport receiver misunderstand the message or cause it to behave in undesirable ways.

- Disclosure. An unauthorized entity may examine the contents of the syslog messages, gaining unauthorized access to the information. Some data in syslog messages is sensitive and may be useful to an attacker, such as the password of an authorized administrator or user.

The secondary threat is:

- Message stream modification. An attacker may delete one or more syslog messages from a series of messages, replay a message, or alter the delivery sequence. The syslog protocol itself is not based on message order. However, an event in a syslog message may relate semantically to events in other messages, so message ordering may be important to understanding a sequence of events.

The following threats are deemed to be of lesser importance for syslog, and are not addressed in this document:

- Denial of Service

- Traffic Analysis

3. Using TLS to Secure Syslog

TLS can be used as a secure transport to counter all the primary threats to syslog described above:

- Confidentiality to counter disclosure of the message contents.

- Integrity-checking to counter modifications to a message on a hop-by-hop basis.

- Server or mutual authentication to counter masquerade.

Note: This secure transport (i.e., TLS) 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, [SYS-SIGN] can be used.

4. Protocol Elements

4.1. Port Assignment

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

The TCP port 6514 has been allocated as the default port for syslog over TLS, as defined in this document.

4.2. Initiation

The transport sender should initiate a connection to the transport receiver and then send the TLS Client Hello to begin the TLS handshake. When the TLS handshake has finished, the transport sender MAY then send the first syslog message.

TLS typically uses certificates [RFC5280] to authenticate peers. Implementations MUST support TLS 1.2 [RFC5246] and are REQUIRED to support the mandatory to implement cipher suite, which is TLS_RSA_WITH_AES_128_CBC_SHA. This document is assumed to apply to future versions of TLS, in which case the mandatory to implement cipher suite for the implemented version MUST be supported.

4.2.1. Certificate-Based Authentication

Both syslog transport sender (TLS client) and syslog transport receiver (TLS 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 TLS. To ensure interoperability between clients and servers, the following methods for certificate validation SHALL be implemented:

- Certification path validation: The TLS peer is configured with one or more trust anchors (typically root CA (certification authority) certificates), which allow it to verify a binding between the subject name and the public key. Additional policy controls needed for authorizing the syslog transport sender and receiver (i.e., verifying that the subject name represents an authorized party) are described in Section 5. Certification path validation is performed as defined in [RFC5280]. This method is useful where there is a Public Key Infrastructure (PKI) deployment.

- End-entity certificate matching: The transport sender or receiver is configured with information necessary to identify the valid end-entity certificates of its authorized peers. The end-entity certificates can be self-signed, and no certification path validation is needed. Implementations MUST support certificate fingerprints in Section 4.2.2 and MAY allow other formats for end-entity certificates such as a DER-encoded certificate. This method provides an alternative to a PKI that is simple to deploy and still maintains a reasonable level of security.

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

The transport receiver and transport sender SHOULD provide mechanisms to record the end-entity certificate for the purpose of correlating it with the sent or received data.

4.2.2. Certificate Fingerprints

Both client and server implementations MUST make the certificate fingerprints for their certificate available through a management interface. The labels for the algorithms are taken from the textual names of the hash functions as defined in the IANA registry "Hash Function Textual Names" allocated in [RFC4572].

The mechanism to generate a fingerprint is to take the hash of the DER-encoded certificate using a cryptographically strong algorithm, and convert the result into colon-separated, hexadecimal bytes, each represented by 2 uppercase ASCII characters. When a fingerprint value is displayed or configured, the fingerprint is prepended with an ASCII label identifying the hash function followed by a colon. Implementations MUST support SHA-1 as the hash algorithm and use the ASCII label "sha-1" to identify the SHA-1 algorithm. The length of a SHA-1 hash is 20 bytes and the length of the corresponding fingerprint string is 65 characters. An example certificate fingerprint is:

```
sha-1:E1:2D:53:2B:7C:6B:8A:29:A2:76:C8:64:36:0B:08:4B:7A:F1:9E:9D
```

During validation the hash is extracted from the fingerprint and compared against the hash calculated over the received certificate.

4.2.3. Cryptographic Level

Syslog applications SHOULD be implemented in a manner that permits administrators, as a matter of local policy, to select the cryptographic level and authentication options they desire.

TLS permits the resumption of an earlier TLS session or the use of another active session when a new session is requested, in order to save the expense of another full TLS handshake. The security parameters of the resumed session are reused for the requested session. The security parameters SHOULD be checked against the security requirements of the requested session to make sure that the resumed session provides proper security.

4.3. Sending Data

All syslog messages MUST be sent as TLS "application data". It is possible for multiple syslog messages to be contained in one TLS record or for a single syslog message to be transferred in multiple TLS 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 protocol [RFC5424].

4.3.1. Message Length

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. 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.

4.4. Closure

A transport sender MUST close the associated TLS connection if the connection is not expected to deliver any syslog messages later. It MUST send a TLS close_notify alert before closing the connection. A transport sender (TLS client) MAY choose to not wait for the transport receiver’s close_notify alert and simply close the connection, thus generating an incomplete close on the transport receiver (TLS server) side. Once the transport receiver gets a close_notify from the transport sender, it MUST reply with a close_notify unless it becomes aware that the connection has already been closed by the transport sender (e.g., the closure was indicated by TCP).

When no data is received from a connection for a long time (where the application decides what "long" means), a transport receiver MAY close the connection. The transport receiver (TLS 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, thus generating an incomplete close on the transport sender side.

5. Security Policies

Different environments have different security requirements and therefore would deploy different security policies. This section discusses some of the security policies that may be implemented by syslog transport receivers and syslog transport senders. The security policies describe the requirements for authentication and authorization. The list of policies in this section is not exhaustive and other policies MAY be implemented.

If the peer does not meet the requirements of the security policy, the TLS handshake MUST be aborted with an appropriate TLS alert.

5.1. End-Entity Certificate Based Authorization

In the simplest case, the transport sender and receiver are configured with information necessary to identity the valid end-entity certificates of its authorized peers.

Implementations MUST support specifying the authorized peers using certificate fingerprints, as described in Section 4.2.1 and Section 4.2.2.

5.2. Subject Name Authorization

Implementations MUST support certification path validation [RFC5280]. In addition, they MUST support specifying the authorized peers using locally configured host names and matching the name against the certificate as follows.

- Implementations MUST support matching the locally configured host name against a dNSName in the subjectAltName extension field and SHOULD support checking the name against the common name portion of the subject distinguished name.

- The '*' (ASCII 42) wildcard character is allowed in the dNSName of the subjectAltName extension (and in common name, if used to store the host name), but only as the left-most (least significant) DNS label in that value. This wildcard matches any left-most DNS label in the server name. That is, the subject *.example.com matches the server names a.example.com and b.example.com, but does not match example.com or a.b.example.com. Implementations MUST support wildcards in certificates as specified above, but MAY provide a configuration option to disable them.

- Locally configured names MAY contain the wildcard character to match a range of values. The types of wildcards supported MAY be more flexible than those allowed in subject names, making it possible to support various policies for different environments. For example, a policy could allow for a trust-root-based authorization where all credentials issued by a particular CA trust root are authorized.

- If the locally configured name is an internationalized domain name, conforming implementations MUST convert it to the ASCII Compatible Encoding (ACE) format for performing comparisons, as specified in Section 7 of [RFC5280].

- Implementations MAY support matching a locally configured IP address against an iPAddress stored in the subjectAltName extension. In this case, the locally configured IP address is converted to an octet string as specified in [RFC5280], Section 4.2.1.6. A match occurs if this octet string is equal to the value of iPAddress in the subjectAltName extension.

5.3. Unauthenticated Transport Sender

In some environments the authenticity of syslog data is not important or is verifiable by other means, so transport receivers may accept data from any transport sender. To achieve this, the transport receiver can skip transport sender authentication (by not requesting client authentication in TLS or by accepting any certificate). In this case, the transport receiver is authenticated and authorized, however this policy does not protect against the threat of transport sender masquerade described in Section 2. The use of this policy is generally NOT RECOMMENDED for this reason.

5.4. Unauthenticated Transport Receiver

In some environments the confidentiality of syslog data is not important, so messages are sent to any transport receiver. To achieve this, the transport sender can skip transport receiver authentication (by accepting any certificate). While this policy does authenticate and authorize the transport sender, it does not protect against the threat of transport receiver masquerade described in Section 2, leaving the data sent vulnerable to disclosure and modification. The use of this policy is generally NOT RECOMMENDED for this reason.

5.5. Unauthenticated Transport Receiver and Sender

In environments where security is not a concern at all, both the transport receiver and transport sender can skip authentication (as described in Sections 5.3 and 5.4). This policy does not protect against any of the threats described in Section 2 and is therefore NOT RECOMMENDED.

6. Security Considerations

This section describes security considerations in addition to those in [RFC5246].

6.1. Authentication and Authorization Policies

Section 5 discusses various security policies that may be deployed. The threats in Section 2 are mitigated only if both the transport sender and transport receiver are properly authenticated and authorized, as described in Sections 5.1 and 5.2. These are the RECOMMENDED configurations for a default policy.

If the transport receiver does not authenticate the transport sender, it may accept data from an attacker. Unless it has another way of authenticating the source of the data, the data should not be trusted. This is especially important if the syslog data is going to be used to detect and react to security incidents. The transport receiver may also increase its vulnerability to denial of service, resource consumption, and other attacks if it does not authenticate the transport sender. Because of the increased vulnerability to attack, this type of configuration is NOT RECOMMENDED.

If the transport sender does not authenticate the syslog transport receiver, then it may send data to an attacker. This may disclose sensitive data within the log information that is useful to an attacker, resulting in further compromises within the system. If a transport sender is operated in this mode, the data sent SHOULD be limited to data that is not valuable to an attacker. In practice this is very difficult to achieve, so this type of configuration is NOT RECOMMENDED.

Forgoing authentication and authorization on both sides allows for man-in-the-middle, masquerade, and other types of attacks that can completely compromise integrity and confidentiality of the data. This type of configuration is NOT RECOMMENDED.

6.2. Name Validation

The subject name authorization policy authorizes the subject in the certificate against a locally configured name. It is generally not appropriate to obtain this name through other means, such as DNS lookup, since this introduces additional security vulnerabilities.

6.3. 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 (or TLS session) 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.

7. IANA Considerations

7.1. Port Number

IANA assigned TCP port number 6514 in the "Registered Port Numbers" range with the keyword "syslog-tls". This port will be the default port for syslog over TLS, as defined in this document.

8. Acknowledgments

Authors appreciate Eric Rescorla, Rainer Gerhards, Tom Petch, Anton Okmianski, Balazs Scheidler, Bert Wijnen, Martin Schuette, Chris Lonvick, and members of the syslog working group for their effort on issues resolving discussion. Authors would also like to thank Balazs Scheidler, Tom Petch, and other persons for their input on security threats of syslog. The authors would like to acknowledge David

Harrington for his detailed reviews of the content and grammar of the document and Pasi Eronen for his contributions to certificate authentication and authorization sections.

9. References

9.1. Normative References

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

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

9.2. Informative References

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, July 2006.

[SYS-SIGN] Kelsey, J., "Signed syslog Messages", Work in Progress, October 2007.

Authors' Addresses

Fuyou Miao (editor)
Huawei Technologies
No. 3, Xinxi Rd
Shangdi Information Industry Base
Haidian District, Beijing 100085
P. R. China

Phone: +86 10 8288 2008
EMail: miaofy@huawei.com
URI: www.huawei.com

Yuzhi Ma (editor)
Huawei Technologies
No. 3, Xinxi Rd
Shangdi Information Industry Base
Haidian District, Beijing 100085
P. R. China

Phone: +86 10 8288 2008
EMail: myz@huawei.com
URI: www.huawei.com

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

EMail: jsalowey@cisco.com