Internet Engineering Task Force (IETF)
Request for Comments: 8253
Updates: 5440
Category: Standards Track
ISSN: 2070-1721

D. Lopez
O. Gonzalez de Dios
Telefonica I+D
Q. Wu
D. Dhody
Huawei

October 2017

PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)

Abstract

The Path Computation Element Communication Protocol (PCEP) defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or among PCEs. This document describes PCEPS -- the usage of Transport Layer Security (TLS) to provide a secure transport for PCEP. The additional security mechanisms are provided by the transport protocol supporting PCEP; therefore, they do not affect the flexibility and extensibility of PCEP.

This document updates RFC 5440 in regards to the PCEP initialization phase procedures.

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 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8253.

Table of Contents

1. Introduction

The Path Computation Element Communication Protocol (PCEP) [RFC5440] defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. These interactions include requests and replies that can be critical for a sustainable network operation and adequate resource allocation; therefore, appropriate security becomes a key element in the PCE infrastructure. As the applications of the PCE framework evolve and more complex service patterns emerge, the definition of a secure mode of operation becomes more relevant.

The Security Considerations section of [RFC5440] analyzes the potential threats to PCEP and their consequences; it also discusses several mechanisms for protecting PCEP against security attacks, without making a specific recommendation on a particular one or defining their application in depth. Moreover, [RFC6952] states the importance of ensuring PCEP communication confidentiality, especially when PCEP communication endpoints do not reside in the same Autonomous System (AS), as the interception of PCEP messages could leak sensitive information related to computed paths and resources.

Transport Layer Security (TLS) [RFC5246] is one of the solutions that seems most adequate among those mentioned in these documents, as it provides support for peer authentication, message encryption, and integrity. TLS provides well-known mechanisms to support key configuration and exchange, as well as means to perform security checks on the results of PCE Discovery (PCED) procedures via the Interior Gateway Protocol (IGP) [RFC5088] [RFC5089].

This document describes a security container for the transport of PCEP messages; therefore, it does not affect the flexibility and extensibility of PCEP.

This document describes how to apply TLS to secure interactions with PCE, including initiation of the TLS procedures, the TLS handshake mechanism, the TLS methods for peer authentication, the applicable TLS ciphersuites for data exchange, and the handling of errors in the security checks. In the rest of this document, we refer to this usage of TLS to provide a secure transport for PCEP as "PCEPS".

Within this document, PCEP communications are described through a PCC-PCE relationship. The PCE architecture also supports PCE-PCE communication; this is achieved by requesting the PCE to fill the role of a PCC, as usual. Thus, in this document, the PCC refers to a PCC or a PCE initiating the PCEP session and acting as a client.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Applying PCEPS

3.1. Overview

The steps involved in establishing a PCEPS session are as follows:

1. Establishment of a TCP connection.

2. Initiation of the TLS procedures by the StartTLS message from PCE to PCC and from PCC to PCE.

3. Negotiation and establishment of a TLS connection.

4. Start exchange of PCEP messages as per [RFC5440].

This document uses the standard StartTLS procedure in PCEP instead of using a different port for the secured session. This is done to avoid requesting allocation of another port number for PCEPS. The StartTLS procedure makes more efficient use of scarce port numbers and allows simpler configuration of PCEP.

Implementations SHOULD follow the best practices and recommendations for using TLS, as per [RFC7525].

It should be noted that this procedure updates what is defined in Sections 4.2.1 and 6.7 of [RFC5440] regarding the initialization phase and the processing of messages prior to the Open message. The details of processing, including backward compatibility, are discussed in the following sections.

3.2. Initiating TLS Procedures

Since PCEP can operate either with or without TLS, it is necessary for a PCEP speaker to indicate whether it wants to set up a TLS connection or not. For this purpose, this document specifies a new PCEP message called "StartTLS". Thus, the PCEP session is secured via TLS from the start, before the exchange of any other PCEP message (including the Open message). This document thus updates [RFC5440], which requires the Open message to be the first PCEP message that is exchanged. In the case of a PCEP session using TLS, the StartTLS

message will be sent first. Also, a PCEP speaker that supports PCEPS MUST NOT start the OpenWait timer after the TCP establishment; instead, it starts a StartTLSWait timer as described in Section 3.3.

The PCEP speaker MAY discover that the PCEP peer supports PCEPS or can be preconfigured to use PCEPS for a given peer (see Section 4 for more details). An existing PCEP session cannot be secured via TLS; the session MUST be closed and re-established with TLS as per the procedure described in this document.

The StartTLS message is a PCEP message sent by a PCC to a PCE and by a PCE to a PCC in order to initiate the TLS procedure for PCEP. The PCC initiates the use of TLS by sending a StartTLS message. The PCE agrees to the use of TLS by responding with its own StartTLS message. If the PCE is configured to only support TLS, it may send the StartTLS message immediately upon TCP connection establishment; otherwise, it MUST wait to see if the PCC's first message is an Open or a StartTLS message. The TLS negotiation and establishment procedures are triggered once the PCEP speaker has sent and received the StartTLS message. The Message-Type field of the PCEP common header for the StartTLS message is set to 13.

Once the TCP connection has been successfully established, the first message sent by the PCC to the PCE and by the PCE to the PCC MUST be a StartTLS message for PCEPS. Note that this is a significant change from [RFC5440], where the first PCEP message is the Open message.

A PCEP speaker receiving a StartTLS message, after any other PCEP exchange has taken place (by receiving or sending any other messages from either side), MUST treat it as an unexpected message and reply with a PCEP Error (PCErr) message with Error-Type set to 25 (PCEP StartTLS failure) and Error-value set to 1 (Reception of StartTLS after any PCEP exchange), and it MUST close the TCP connection.

Any message received prior to the StartTLS or Open message MUST trigger a protocol error condition causing a PCErr message to be sent with Error-Type set to 25 (PCEP StartTLS failure) and Error-value set to 2 (Reception of any other message apart from StartTLS, Open, or PCErr), and it MUST close the TCP connection.

If the PCEP speaker that does not support PCEPS receives a StartTLS message, it will behave according to the existing error mechanism described in Section 6.2 of [RFC5440] (if the message is received prior to an Open message) or Section 6.9 of [RFC5440] (if an unknown message is received). See Section 5 for more details.

If the PCEP speaker that only supports PCEPS connections (as a local policy) receives an Open message, it MUST treat it as an unexpected message and reply with a PCErr message with Error-Type set to 1 (PCEP session establishment failure) and Error-value set to 1 (reception of an invalid Open message or a non Open message), and it MUST close the TCP connection.

If a PCC supports PCEPS connections and allows non-PCEPS connections (as a local policy), it MUST first try to establish PCEPS by sending a StartTLS message, and in case it receives a PCErr message from the PCE, it MAY retry to establish a connection without PCEPS by sending an Open message. If a PCE supports PCEPS connections and allows non-PCEPS connections (as a local policy), it MUST wait to respond after TCP establishment, based on the message received from the PCC. In case of a StartTLS message, the PCE MUST respond by sending a StartTLS message and moving to TLS establishment procedures as described in this document. In case of an Open message, the PCE MUST respond with an Open message and move to the PCEP session establishment procedure as per [RFC5440]. If a PCE supports PCEPS connections only (as a local policy), it MAY send a StartTLS message to the PCC without waiting to receive a StartTLS message from the PCC.

If a PCEP speaker that is unwilling or unable to negotiate TLS receives a StartTLS message, it MUST return a PCErr message (in the clear) with Error-Type set to 25 (PCEP StartTLS failure) and Error-value set to:

- 3 (Failure, connection without TLS is not possible) if it is not willing to exchange PCEP messages without the solicited TLS connection, and it MUST close the TCP session.

- 4 (Failure, connection without TLS is possible) if it is willing to exchange PCEP messages without the solicited TLS connection, and it MUST close the TCP session. The receiver MAY choose to attempt to re-establish the PCEP session without TLS next. Re-establishing the PCEP session without TLS SHOULD be limited to only one attempt.

If the PCEP speaker supports PCEPS and can establish a TLS connection, it MUST start the TLS connection negotiation and establishment steps described in Section 3.4 before the PCEP initialization procedure (see Section 4.2.1 of [RFC5440]).

After the exchange of StartTLS messages, if the TLS negotiation fails for some reason (e.g., the required mechanisms for certificate revocation checking are not available), both peers MUST immediately close the connection.

A PCEP speaker that does not support PCEPS sends the Open message directly, as per [RFC5440]. A PCEP speaker that supports PCEPS, but has learned in the last exchange the peer's willingness to re-establish the session without TLS, MAY send the Open message directly, as per [RFC5440]. Re-establishing the PCEP session without TLS SHOULD be limited to only one attempt.

Given the asymmetric nature of TLS for connection establishment, it is relevant to identify the roles of each of the PCEP peers in it. The PCC SHALL act as the TLS client, and the PCE SHALL act as the TLS server as per [RFC5246].

As per the recommendation from [RFC7525] to avoid downgrade attacks, PCEP peers that support PCEPS SHOULD default to strict TLS configuration, i.e., not allowing non-TLS PCEP sessions to be established. PCEPS implementations MAY provide an option to allow the operator to manually override strict TLS configuration and allow unsecured connections. Execution of this override SHOULD trigger a warning about the security implications of permitting unsecured connections.

3.3. The StartTLS Message

The StartTLS message is used to initiate the TLS procedure for a PCEPS session between the PCEP peers. A PCEP speaker sends the StartTLS message to request negotiation and establishment of a TLS connection for PCEP. On receiving a StartTLS message from the PCEP peer (i.e., when the PCEP speaker has sent and received the StartTLS message), it is ready to start the negotiation and establishment of TLS and move to the steps described in Section 3.4.

The collision resolution procedures described in [RFC5440] for the exchange of Open messages MUST be applied by the PCEP peers during the exchange of StartTLS messages.

The format of a StartTLS message is as follows:

```
<StartTLS Message>::= <Common Header>
```

The StartTLS message MUST contain only the PCEP common header with the Message-Type field set to 13.

Once the TCP connection has been successfully established, the PCEP speaker MUST start a timer called the "StartTLSWait timer". After the expiration of this timer, if neither the StartTLS message nor a PCErr/Open message (in case of failure and PCEPS not being supported by the peer, respectively) has been received, the PCEP speaker MUST send a PCErr message with Error-Type set to 25 (PCEP StartTLS

failure) and Error-value set to 5 (No StartTLS message (nor PCErr/ Open) before StartTLSWait timer expiry), and it MUST release the TCP connection. A RECOMMENDED value for the StartTLSWait timer is 60 seconds. The value of the StartTLSWait timer MUST NOT be less than that of the OpenWait timer.

The following figures illustrate the various interactions between a PCC and a PCE, based on the support for the PCEPS capability, during the PCEP session initialization.

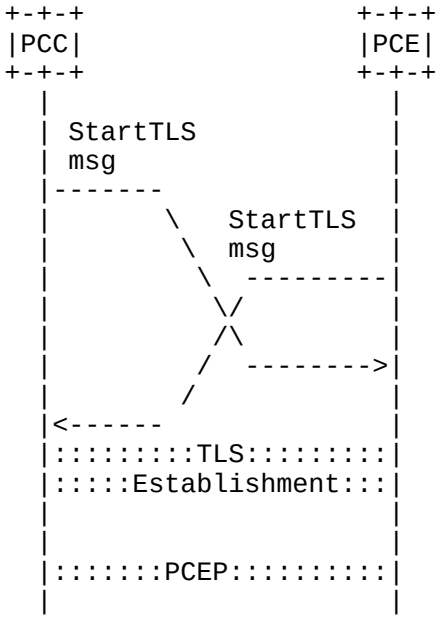

Figure 1: Both PCEP speakers support PCEPS (strict)

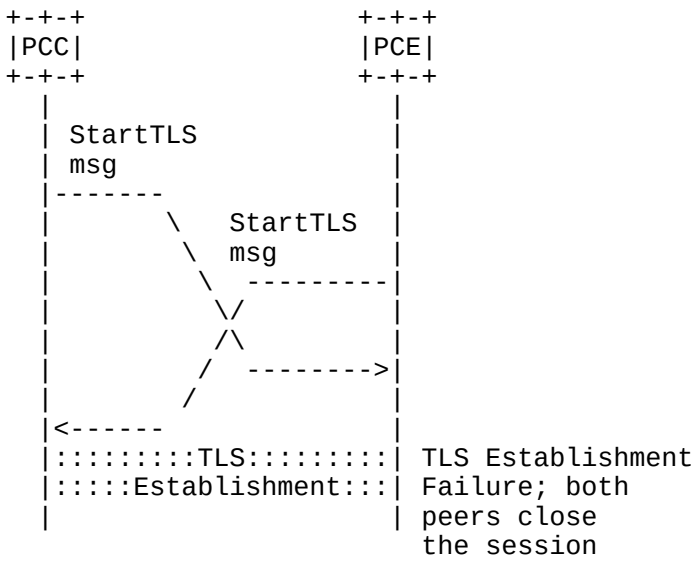

Figure 2: Both PCEP speakers support PCEPS (strict) but cannot establish TLS

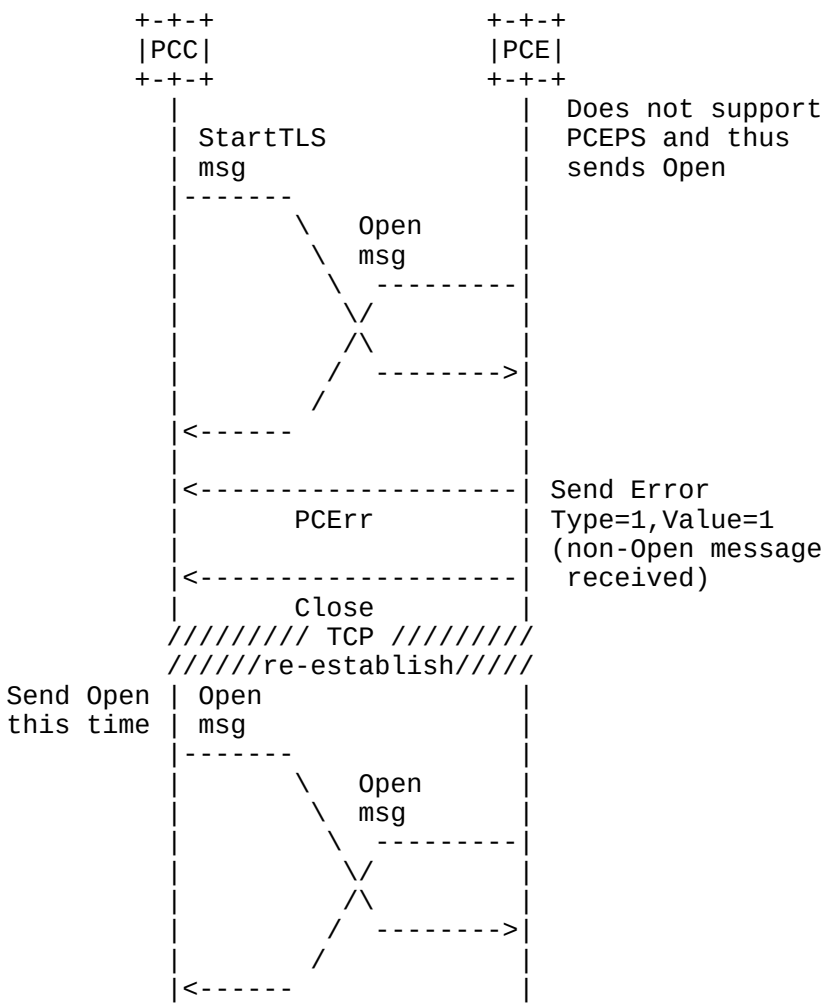

Figure 3: PCE does not support connection with PCEPS, whereas PCC supports connection with or without PCEPS

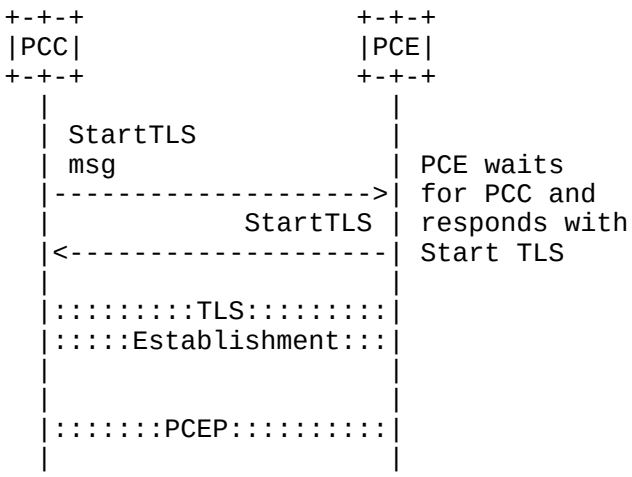

Figure 4: Both PCEP speakers support connection with or without PCEPS

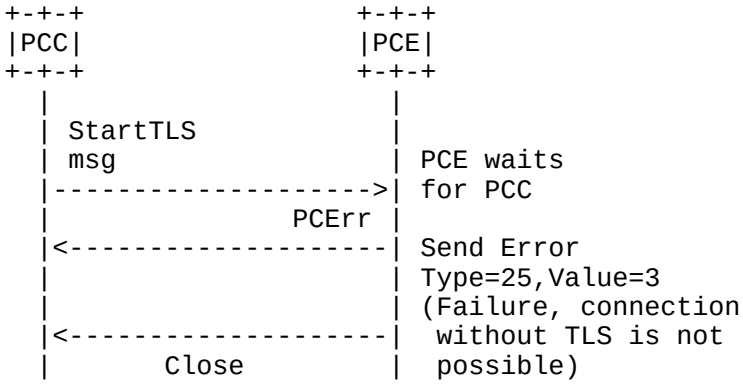

Figure 5: Both PCEP speakers support connection with or without PCEPS, but PCE cannot start TLS negotiation

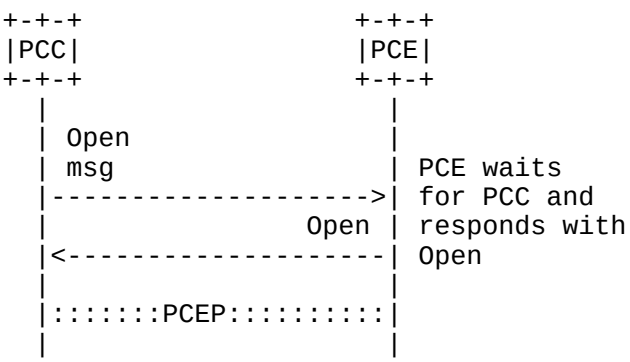

Figure 6: PCE supports connection with or without PCEPS, whereas PCC does not support connection with PCEPS

3.4. TLS Connection Establishment

Once the establishment of TLS has been agreed upon by the PCEP peers, the connection establishment SHALL follow the following steps:

1. Immediately negotiate a TLS session according to [RFC5246]. The following restrictions apply:

   * Support for TLS v1.2 [RFC5246] or later is REQUIRED.

   * Support for certificate-based mutual authentication is REQUIRED.

   * Negotiation of a ciphersuite providing for integrity protection is REQUIRED.

   * Negotiation of a ciphersuite providing for confidentiality is RECOMMENDED.

   * Support for and negotiation of compression is OPTIONAL.

   * PCEPS implementations MUST, at a minimum, support negotiation of the TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 [RFC6460] and SHOULD support TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as well. Implementations SHOULD support the NIST P-256 (secp256r1) curve [RFC4492]. In addition, PCEPS implementations MUST support negotiation of the mandatory-to-implement ciphersuites required by the versions of TLS that they support from TLS 1.3 onwards.

2. Peer authentication can be performed in any of the following two REQUIRED operation models:

   * TLS with X.509 certificates using Public-Key Infrastructure Exchange (PKIX) trust models:

     + Implementations MUST allow the configuration of a list of trusted Certification Authorities (CAs) for incoming connections.

     + Certificate validation MUST include the verification rules as per [RFC5280].

     + PCEPS implementations SHOULD incorporate revocation methods (Certificate Revocation List (CRL) downloading, Online Certificate Status Protocol (OCSP), etc.) according to the trusted CA policies.

     + Implementations SHOULD indicate their trusted CAs. For TLS 1.2, this is done using "certificate_authorities" on the server side (see Section 7.4.4 of [RFC5246]) and the "TrustedAuthorities" extension on the client side (see Section 6 of [RFC6066]).

     + Implementations MUST follow the rules and guidelines for peer validation as defined in [RFC6125]. If an expected DNS name or IP address for the peer is configured, then the implementations MUST check them against the values in the presented certificate. The DNS names and the IP addresses can be contained in the Common Name Identifier (CN-ID) [RFC6125] or the subjectAltName entries. For verification, only one of these entries is considered. The following precedence applies: for DNS name validation, DNS-ID [RFC6125] has precedence over CN-ID, and for IP address validation, subjectAltName:iPAddr has precedence over CN-ID.

     + Implementations MAY allow the configuration of a set of additional properties of the certificate to check for a peer’s authorization to communicate (e.g., a set of allowed values in URI-ID [RFC6125] or a set of allowed X.509 v3 Certificate Policies). The definitions of these properties are out of scope of this document.

   * TLS with X.509 certificates using certificate fingerprints: Implementations MUST allow the configuration of a list of certificates that are trusted to identify peers, identified via the fingerprint of certificate octets encoded by the

Distinguished Encoding Rules (DER). Implementations MUST support SHA-256 as defined by [SHS] as the hash algorithm for the fingerprint, but a later revision may demand support for a stronger hash function.

3. Start exchanging PCEP messages.

   * Once the TLS connection has been successfully established, the PCEP speaker MUST start the OpenWait timer [RFC5440]; after the expiration of this timer, if no Open message has been received, the PCEP speaker sends a PCErr message and releases the TCP/TLS connection.

3.5. Peer Identity

Depending on the peer authentication method in use, PCEPS supports different operation modes to establish a peer's identity and whether it is entitled to perform requests or can be considered authoritative in its replies. PCEPS implementations SHOULD provide mechanisms for associating peer identities with different levels of access and/or authoritativeness, and they MUST provide a mechanism for establishing a default level for properly identified peers. Any connection established with a peer that cannot be properly identified SHALL be terminated before any PCEP exchange takes place.

In TLS X.509 mode using fingerprints, a peer is uniquely identified by the fingerprint of the presented certificate.

There are numerous trust models in PKIX environments, and it is beyond the scope of this document to define how a particular deployment determines whether a peer is trustworthy. Implementations that want to support a wide variety of trust models should expose as many details of the presented certificate to the administrator as possible so that the trust model can be implemented by the administrator. At least the following parameters of the X.509 certificate SHOULD be exposed:

- Peer's IP Address
- Peer's Fully Qualified Domain Name (FQDN)
- Certificate Fingerprint
- Issuer
- Subject
- All X.509 v3 Extended Key Usage

- All X.509 v3 Subject Alternative Name

- All X.509 v3 Certificate Policies

Note that the remote IP address used for the TCP session establishment is also exposed.

[RFC8232] specifies a Speaker Entity Identifier TLV (SPEAKER-ENTITY-ID) as an optional TLV that is included in the OPEN object. It contains a unique identifier for the node that does not change during the lifetime of the PCEP speaker. An implementation would thus expose the speaker entity identifier as part of the X.509 v3 certificate's subjectAltName:otherName, so that an implementation could use this identifier for the peer identification trust model.

In addition, a PCC MAY apply the procedures described in "DNS-Based Authentication of Named Entities (DANE)" [RFC6698] to verify its peer identity when using DNS discovery. See Section 4.1 for further details.

3.6. Connection Establishment Failure

In case the initial TLS negotiation or the peer identity check fails, according to the procedures listed in this document, both peers MUST immediately close the connection.

The initiator SHOULD follow the procedure listed in [RFC5440] to retry session setup as per the exponential back-off session establishment retry procedure.

4. Discovery Mechanisms

This document does not specify any discovery mechanism for support of PCEPS. [PCE-DISCOVERY-PCEPS-SUPPORT] and [PCE-DISCOVERY-DNS] make the following proposals:

- A PCE can advertise its capability to support PCEPS using the IGP's advertisement mechanism of the PCED information. The PCE-CAP-FLAGS sub-TLV is an optional sub-TLV used to advertise PCE capabilities. It is present within the PCED sub-TLV carried by OSPF or IS-IS. [RFC5088] and [RFC5089] provide the description and processing rules for this sub-TLV when carried within OSPF and IS-IS, respectively. PCE capability bits are defined in [RFC5088]. A new capability flag bit for the PCE-CAP-FLAGS sub-TLV that can be announced as an attribute to distribute PCEP security support information is proposed in [PCE-DISCOVERY-PCEPS-SUPPORT].

- o A PCE can advertise its capability to support PCEPS using DNS [PCE-DISCOVERY-DNS] by identifying the support of TLS.

4.1. DANE Applicability

DANE [RFC6698] defines a secure method to associate the certificate that is obtained from a TLS server with a domain name using DNS, i.e., using the TLSA DNS resource record (RR) to associate a TLS server certificate or public key with the domain name where the record is found, thus forming a "TLSA certificate association". The DNS information needs to be protected by DNS Security (DNSSEC). A PCC willing to apply DANE to verify server identity MUST conform to the rules defined in Section 4 of [RFC6698]. The implementation MUST support service certificate constraint (TLSA certificate usages type 1) with Matching type 1 (SHA2-256) as described in [RFC6698] and [RFC7671]. The server's domain name must be authorized separately, as TLSA does not provide any useful authorization guarantees.

5. Backward Compatibility

The procedures described in this document define a security container for the transport of PCEP requests and replies carried by a TLS connection initiated by means of a specific extended message (StartTLS) that does not interfere with PCEP speaker implementations not supporting it.

A PCC that does not support PCEPS will send an Open message as the first message on TCP establishment. A PCE that only supports PCEPS will send a StartTLS message on TCP establishment. The PCC would consider the received StartTLS message as an error and behave according to the existing error mechanism of [RFC5440], i.e., it would send a PCErr message with Error-Type 1 (PCEP session establishment failure) and Error-value 1 (reception of an invalid Open message or a non Open message) and close the session.

A PCC that support PCEPS will send a StartTLS message as the first message on TCP establishment. A PCE that does not support PCEPS would consider receiving a StartTLS message as an error, respond with a PCErr message with Error-Type 1 (PCEP session establishment failure) and Error-value 1 (reception of an invalid Open message or a non Open message), and close the session.

If a StartTLS message is received at any other time by a PCEP speaker that does not implement PCEPS, it would consider it as an unknown message and would behave according to the existing error mechanism of [RFC5440], i.e., it would send a PCErr message with Error-Type 2 (Capability not supported) and close the session.

An existing PCEP session cannot be upgraded to PCEPS; the session needs to be terminated and re-established as per the procedure described in this document. During the incremental upgrade, the PCEP speaker SHOULD allow session establishment with and without TLS. Once both PCEP speakers are upgraded to support PCEPS, the PCEP session is re-established with TLS; otherwise, a PCEP session without TLS is set up. A redundant PCE MAY also be used during the incremental deployment to take over the PCE undergoing upgrade. Once the upgrade is completed, support for the unsecured version SHOULD be removed.

A PCE that accepts connections with or without PCEPS would respond based on the message received from the PCC. A PCC that supports connection with or without PCEPS would first attempt to connect with PCEPS, and in case of error, it MAY retry to establish connection without PCEPS. For successful TLS operations with PCEP, both PCEP peers in the network would need to be upgraded to support this document.

Note that a PCEP implementation that supports PCEPS would respond with a PCErr message with Error-Type set to 25 (PCEP StartTLS failure) and Error-value set to 2 (Reception of any other message apart from StartTLS, Open, or PCErr) if any other message is sent before a StartTLS or Open message. If the sender of the invalid message is a PCEP implementation that does not support PCEPS, it will not be able to understand this error. A PCEPS implementation could also send the PCErr message as per [RFC5440] with Error-Type 1 (PCEP session establishment failure) and Error-value 1 (reception of an invalid Open message or a non Open message) before closing the session.

6. IANA Considerations

6.1. New PCEP Message

The following new message type has been allocated within the "PCEP Messages" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry:

Value	Description	Reference
13	StartTLS	This document

6.2. New Error-Values

The following new error types and error values have been allocated within the "PCEP-ERROR Object Error Types and Values" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry:

Error-Type	Meaning	Error-value	Reference
25	PCEP StartTLS failure	0: Unassigned	This document
		1: Reception of StartTLS after any PCEP exchange	This document
		2: Reception of any other message apart from StartTLS, Open, or PCErr	This document
		3: Failure, connection without TLS is not possible	This document
		4: Failure, connection without TLS is possible	This document
		5: No StartTLS message (nor PCErr/Open) before StartTLSWait timer expiry	This document

7. Security Considerations

While the application of TLS satisfies the requirement on confidentiality as well as fine-grained, policy-based peer authentication, there are security threats that it cannot address. It may be advisable to apply additional protection measures, in particular in what relates to attacks specifically addressed to forging the TCP connection underpinning TLS, especially in the case of long-lived connections. One of these measures is the application of the TCP Authentication Option (TCP-AO) [RFC5925], which is fully compatible with and deemed as complementary to TLS. The mechanisms to configure the requirements to use TCP-AO and other lower-layer protection measures with a particular peer are outside the scope of this document.

Since computational resources required by the TLS handshake and ciphersuite are higher than unencrypted TCP, clients connecting to a PCEPS server can more easily create high-load conditions, and a malicious client might create a denial-of-service attack more easily.

Some TLS ciphersuites only provide integrity validation of their payload and provide no encryption; such ciphersuites SHOULD NOT be used by default. Administrators MAY allow the usage of these ciphersuites after careful weighting of the risk of relevant internal data leakage that can occur in such a case, as explicitly stated by [RFC6952].

When using certificate fingerprints to identify PCEPS peers, any two certificates that produce the same hash value will be considered the same peer. Therefore, it is important to make sure that the hash function used is cryptographically uncompromised, so that attackers are very unlikely to be able to produce a hash collision with a certificate of their choice. This document mandates support for SHA-256 as defined by [SHS], but a later revision may demand support for stronger functions if suitable attacks on it are known.

PCEPS implementations that continue to accept connections without TLS are susceptible to downgrade attacks as described in [RFC7457]. An attacker could attempt to remove the use of StartTLS messages that request the use of TLS as it pass on the wire in clear and could also attempt to inject a PCErr message that suggests attempting PCEP connection without TLS.

The guidance given in [RFC7525] SHOULD be followed to avoid attacks on TLS.

8. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

8.1. Control of Function and Policy

A PCE or PCC implementation SHOULD allow configuring the PCEP security via TLS capabilities as described in this document.

A PCE or PCC implementation supporting PCEP security via TLS MUST support general TLS configuration as per [RFC5246]. At least the configuration of one of the trust models and its corresponding parameters, as described in Sections 3.4 and 3.5, MUST be supported by the implementation.

A PCEPS implementation SHOULD allow configuring the StartTLSWait timer value.

PCEPS implementations MAY provide an option to allow the operator to manually override strict TLS configuration and allow unsecure connections. Execution of this override SHOULD trigger a warning about the security implications of permitting unsecure connections.

Further, the operator needs to develop suitable security policies around PCEP within his network. The PCEP peers SHOULD provide ways for the operator to complete the following tasks in regards to a PCEP session:

- Determine if a session is protected via PCEPS.
- Determine the version of TLS, the mechanism used for authentication, and the ciphersuite in use.
- Determine if the certificate could not be verified and the reason for this circumstance.
- Inspect the certificate offered by the PCEP peer.
- Be warned if the StartTLS procedure fails for the PCEP peers that are known to support PCEPS via configurations or capability advertisements.

8.2. Information and Data Models

The PCEP MIB module is defined in [RFC7420]. The MIB module could be extended to include the ability to view the PCEPS capability, TLS-related information, and the TLS status for each PCEP peer.

Further, to allow the operator to configure the PCEPS capability and various TLS-related parameters as well as to view the current TLS status for a PCEP session, the PCEP YANG module [PCEP-YANG] is extended to include TLS-related information.

8.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440] and [RFC5246].

8.4. Verifying Correct Operations

A PCEPS implementation SHOULD log error events and provide PCEPS failure statistics with reasons.

8.5. Requirements on Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols. Note that Section 4 lists possible discovery mechanisms for support of PCEPS.

8.6. Impact on Network Operation

Mechanisms defined in this document do not have any significant impact on network operations in addition to those already listed in [RFC5440] and on the policy and management implications discussed above.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671, October 2015, <https://www.rfc-editor.org/info/rfc7671>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[SHS] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS PUB 180-4, DOI 10.6028/NIST.FIPS.180-4, August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

9.2. Informative References

[PCE-DISCOVERY-DNS]
Wu, Q., Dhody, D., King, D., Lopez, D., and J. Tantsura, "Path Computation Element (PCE) Discovery using Domain Name System(DNS)", Work in Progress, draft-wu-pce-dns-pce-discovery-10, March 2017.

[PCE-DISCOVERY-PCEPS-SUPPORT]
Lopez, D., Wu, Q., Dhody, D., Wang, Z., and D. King, "IGP extension for PCEP security capability support in the PCE discovery", Work in Progress, draft-wu-pce-discovery-pceps-support-07, March 2017.

[PCEP-YANG] Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", Work in Progress, draft-ietf-pce-pcep-yang-05, July 2017.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <https://www.rfc-editor.org/info/rfc4492>.

[RFC4513] Harrison, R., Ed., "Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms", RFC 4513, DOI 10.17487/RFC4513, June 2006, <https://www.rfc-editor.org/info/rfc4513>.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <https://www.rfc-editor.org/info/rfc5088>.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <https://www.rfc-editor.org/info/rfc5089>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC6460] Salter, M. and R. Housley, "Suite B Profile for Transport Layer Security (TLS)", RFC 6460, DOI 10.17487/RFC6460, January 2012, <https://www.rfc-editor.org/info/rfc6460>.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <https://www.rfc-editor.org/info/rfc6614>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <https://www.rfc-editor.org/info/rfc6952>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <https://www.rfc-editor.org/info/rfc7420>.

[RFC7457] Sheffer, Y., Holz, R., and P. Saint-Andre, "Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)", RFC 7457, DOI 10.17487/RFC7457, February 2015, <https://www.rfc-editor.org/info/rfc7457>.

[RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <https://www.rfc-editor.org/info/rfc8232>.

Acknowledgements

This specification relies on the analysis and profiling of TLS included in [RFC6614] and the procedures described for the StartTLS command in [RFC4513].

We would like to thank Joe Touch for his suggestions and support regarding the StartTLS mechanisms.

Thanks to Daniel King for reminding the authors about manageability considerations.

Thanks to Cyril Margaria for shepherding this document.

Thanks to David Mandelberg for early SECDIR review comments as well as further review during IETF last call.

Thanks to Dan Frost for the RTGDIR review and comments.

Thanks to Dale Worley for the Gen-ART review and comments.

Thanks to Tianran Zhou for the OPSDIR review.

Thanks to Deborah Brungard for being the responsible AD and guiding the authors as needed.

Also, thanks to Mirja Kuhlewind, Eric Rescorla, Warren Kumari, Kathleen Moriarty, Suresh Krishnan, Ben Campbell, and Alexey Melnikov for the IESG review and comments.

Authors' Addresses

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
Email: diego.r.lopez@telefonica.com

Oscar Gonzalez de Dios
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
Email: oscar.gonzalezdedios@telefonica.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

Dhruv Dhody
Huawei
Divyashree Techno Park, Whitefield
Bangalore, KA 560066
India

Email: dhruv.ietf@gmail.com