Internet Engineering Task Force (IETF)
Request for Comments: 6876
Category: Standards Track
ISSN: 2070-1721

P. Sangster
Symantec Corporation
N. Cam-Winget
J. Salowey
Cisco Systems

February 2013

A Posture Transport Protocol over TLS (PT-TLS)

Abstract

This document specifies PT-TLS, a TLS-based Posture Transport (PT) protocol. The PT-TLS protocol carries the Network Endpoint Assessment (NEA) message exchange under the protection of a Transport Layer Security (TLS) secured tunnel.

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/rfc6876.

Table of Contents

1. Introduction

The NEA architecture [RFC5209] defines a system for communicating posture between a client, where it is collected, and server, where it is assessed. Posture is configuration and/or status of hardware or software on an endpoint as it pertains to an organization's security policy. This document specifies PT-TLS, a TLS-based Posture Transport (PT) protocol protected by a TLS channel.

NEA protocols are intended to be used for pre-admission assessment of endpoints joining the network and to assess endpoints already present on the network. In order to support both usage models, two different types (or bindings) of PT protocols are necessary to operate before and after the endpoint has an assigned IP address and other network-layer information. This specification focuses on the PT protocol used to assess endpoints already present on the network and thus is able to use TCP/IP-based transport protocols. NEA has defined another protocol called PT-EAP [PT-EAP] to address assessment prior to the endpoint having an assigned IP address.

The Posture Transport protocol in the NEA architecture [RFC5209] is responsible for transporting Posture Broker (PB-TNC [RFC5793]) batches, often containing Posture Attributes (PA-TNC [RFC5792]) over the network between the Posture Transport Client component of the NEA Client and the Posture Transport Server component of the NEA Server.

The PT protocol also offers strong security protections to ensure that the exchanged messages are protected from a variety of threats from hostile intermediaries.

1.1. Prerequisites

This document does not define an architecture or reference model. Instead, it defines one binding of the PT protocol that works within the reference model described in the NEA Overview and Requirements specification [RFC5209]. The reader is assumed to be thoroughly familiar with [RFC5209]. The NEA working group compared the functionality described in this specification with the requirements in [RFC5209] and found that each applicable requirement was addressed.

1.2. Message Diagram Conventions

This specification defines the syntax of PT-TLS messages using diagrams. Each diagram depicts the format and size of each field in bits. Implementations MUST send the bits in each diagram as they are shown, traversing the diagram from top to bottom and then from left to right within each line (which represents a 32-bit quantity). Multi-byte fields representing numeric values must be sent in network (big endian) byte order.

Bit field (e.g., flag) values are described referring to the position of the bit within the field. These bit positions are numbered from the most significant bit through the least significant bit, so a one-octet field with only bit 0 set has the value 0x80.

1.3. Conventions Used in This Document

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 RFC 2119 [RFC2119].

1.4. Compatibility with Other Specifications

One of the goals of the NEA effort is to deliver a single set of endpoint assessment standards, agreed upon by all parties. For this reason, the authors understand that the Trusted Computing Group (TCG) will be replacing its existing posture transport protocols with new versions that are equivalent to and interoperable with the NEA specifications.

2. Design Considerations

This section discusses some of the key design considerations for the PT protocol. This document specifies the PT binding for use when performing an assessment or reassessment after the endpoint has been admitted to the network and is capable of using TCP/IP to communicate with the NEA Server. If the endpoint does not yet have TCP/IP-layer access to the NEA Server (and vice versa), the endpoint can use the PT-EAP (Posture Transport (PT) Protocol for Extensible Authentication Protocol (EAP) Tunnel Methods) protocol when performing an assessment.

Because the endpoint has TCP/IP access to the NEA Server (potentially on a restricted portion of the network), the NEA Client and NEA Server have the ability to establish (or re-use) a reliable TCP/IP connection in order to perform the assessment. The TCP/IP connection enables the assessment to occur over a relatively high-performance, reliable channel capable of supporting multiple roundtrip message exchanges in a full-duplex manner. These connection properties are very different from what is available when the endpoint is initially joining the network (e.g., during an 802.1X-based assessment); therefore, the design described in this specification follows a different path to maximize the benefits of the underlying TCP/IP connection.

2.1. Benefits of TCP/IP Connectivity

The PT protocol over TLS is typically able to offer to the NEA Client and NEA Server significantly higher quality of service and flexibility of operation than PT-EAP. However, there may be some added risks when the endpoint is on the network prior to its initial assessment (if no admission time assessment had been performed). Because of these risks, the combined use of an EAP-based assessment during admission followed by reassessment using TCP/IP may be appropriate in some environments.

Some of the benefits to having a TCP/IP-based transport during an assessment include:

- Full-Duplex Connectivity - used to support asynchronous initiation of posture exchanges within a single TLS connection (e.g., triggered by alerts of posture or policy changes)

- High Bandwidth - potentially much higher bandwidth than other transports (e.g., EAP), allowing more in-band data (e.g., remediation, verbose posture information)

- Large Messages - ability to send very large Posture Attribute messages without directly fragmenting them (underlying carrier protocol may introduce fragmentation)

- Bidirectional - NEA Client and NEA Server can initiate an assessment or reassessment

- Multiple Roundtrips - NEA Client and NEA Server can exchange numerous messages without fear of infrastructure timeouts. However, the entire exchange should be kept as brief as possible if the user has to wait for its completion.

2.2. Leveraging Proven TLS Security

All PT protocol bindings must be capable of providing strong authentication, integrity, and confidentiality protection for the PB-TNC batches. Rather than define a new protocol over TCP/IP to provide adequate protection, this specification requires the use of Transport Layer Security [RFC5246] to secure the connection. TLS was selected because it's a widely deployed protocol with parallel protections to a number of the EAP tunnel methods, and it meets all of the security requirements.

2.3. TLV-Based Message Encapsulation

The design of the PT-TLS protocol is based upon the use of a type-length-value (TLV)-oriented protocol message that identifies the type of message, the message's length, and a potentially variable-length payload value. The use of a TLV-oriented encoding was chosen to match the Internet standard PA-TNC and PB-TNC protocols. Because the PA-TNC, PB-TNC, and PT-TLS protocols are typically implemented inside the same process space, this allows a common set of message-parsing code to be used. Similarly, creation of debugging tools is simplified by the common encoding methodologies. TLV-based encoding was used in each of the NEA protocols in part because it enables a very space-efficient representation on the network and is simpler to parse than some other encodings to benefit lower-powered (or battery constrained) devices.

2.4. No Change to Base TLS Protocol

During the design of the PT-TLS protocol, several approaches were considered with different costs and benefits. Several considered approaches involved integrating the PT protocol into the TLS handshake protocol. Because the PT protocol requires the underlying TLS carrier to provide security protections, the PT protocol couldn't operate before the cipher suites were negotiated and in use. One option was to integrate into the TLS handshake protocol after the

ChangeCipherSpec phase, allowing the PT message to be protected. The benefit of this approach is that the assessment protocol could operate below the application protocols, allowing for easier integration into applications. However, making this change would require some extensions to the TLS handshake protocol standards and existing widely deployed TLS implementations, so it wasn't clear that the cost was warranted, particularly because the application independence can also be offered by a shim library between the application and TLS library that provides the PT protocol encapsulation/decapsulation.

The other general approach considered was to have PT-TLS layer on top of TLS as an application protocol (using the standard application_data ContentType). This has the advantage that existing TLS software could be used. However, the PB-TNC traffic would need to be encapsulated/decapsulated by a new PT-TLS protocol layer before being passed to the TLS library. This didn't seem like a significant issue, as PB-TNC is architected to layer on PT anyway.

After considering the different options, it was determined that layering the PT protocol on top of the TLS protocol without requiring current TLS protocol implementations to change met all the requirements and offered the best path toward rapid adoption and deployment. Therefore, the following sections describe a PT protocol that is carried on top of TLS.

3. PT-TLS Protocol

This section specifies the PT-TLS protocol, a Posture Transport (PT) protocol carried by the Transport Layer Security (TLS) protocol over a TCP/IP network. As shown in Figure 1, this protocol runs directly on top of TLS as an application. This means PT-TLS is encapsulated within the TLS Record Layer protocol using the standard ContentType for applications (application_data).

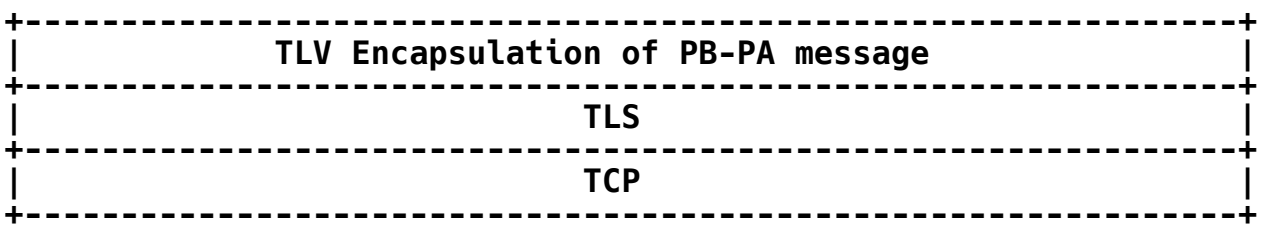


Figure 1. PT-TLS Layering Model

3.1. Initiating a PT-TLS Session

The PT-TLS protocol may be initiated by a Posture Transport Client or a Posture Transport Server. This flexibility supports different use cases. For example, a Posture Transport Client that wishes to trigger a NEA assessment to determine whether its security posture is good can start up a PT-TLS session and request a posture assessment. On the other hand, when an endpoint requests access to a protected network or resource, a Posture Transport Server can start up a PT-TLS session and perform a posture assessment before deciding whether to grant access.

The party that initiates a PT-TLS session is known as the "PT-TLS Initiator". The other party in the session (which receives the request to open a PT-TLS session) is known as the "PT-TLS Responder".

3.1.1. Issues with Server-Initiated PT-TLS Sessions

In order for a NEA Server to establish a PT-TLS session, the NEA Client needs to be listening for a connection request on a TCP port known by the NEA Server. In many deployments, the security policies of an endpoint (e.g., firewall software) or the security policies of a network (e.g., firewall devices) are designed to minimize the number of open inbound TCP/UDP ports that are available to the network to reduce the potential attack footprint. This is one issue that makes it difficult for a NEA Server to initiate a PT-TLS session.

Another issue with this scenario involves X.509 certificates. When the NEA Server creates a TLS session to the NEA Client, the NEA Client is effectively acting as the TLS server during the TLS protocol exchange. This means the NEA Client would typically need to possess an X.509 certificate to protect the initial portion of the TLS handshake. In situations where the NEA Server initiates the creation of the TLS session, both the NEA Client and NEA Server MUST possess X.509 certificates to fully authenticate the session. For many deployments, provisioning X.509 certificates to all NEA Clients has scalability and cost issues; therefore, it is recommended that the NEA Client not listen for connection requests from the NEA Server but instead establish and maintain a TLS session to the NEA Server proactively, so either party can initiate an assessment using the preexisting TLS session as required.

In most cases, the traditional methods of server certificate ID validation will not apply when the NEA Server initiates the connection. In this case, the NEA Client and Server need to follow the certificate path validation rules in RFC 5280 [RFC5280]. In

addition, each side needs to be able to authorize its peer based upon matching Subject and SubjectAltName fields for certificates issued by a particular trust anchor.

Therefore, NEA Clients SHOULD be capable of establishing and holding open a TLS session with the NEA Server immediately after obtaining network access. A NEA Client MAY listen for connection requests from the NEA Server and establish a new PT-TLS session when one does not already exist. Because of the potential added complexity, a NEA Client's support for accepting inbound PT-TLS connections is optional to implement. Having an existing PT-TLS session allows either party to initiate an assessment without requiring the NEA Client to be listening for new connection requests. In order to keep the TLS session alive, the NEA Client and NEA Server SHOULD be capable of supporting the TLS heartbeat protocol [RFC6520].

3.1.2. Establish or Re-Use Existing PT-TLS Session

A single PT-TLS session can support multiple NEA assessments, which can be started by either party (the PT-TLS Initiator or the PT-TLS Responder). The party that starts a NEA assessment is known as the "assessment initiator", and the other party is known as the "assessment responder".

If the assessment initiator already has a PT-TLS session to the assessment responder, the initiator can re-use this session; otherwise, a new PT-TLS session needs to be established.

3.2. TCP Port Usage

In order for a PT-TLS Initiator to establish a TCP connection to a PT-TLS Responder, the initiator needs to know the TCP port number on which the responder is listening for assessment requests. The IANA has reserved TCP port number 271 for use by "pt-tls".

3.3. Preventing MITM Attacks with Channel Bindings

As described in "The Network Endpoint Assessment (NEA) Asokan Attack Analysis" [RFC6813], a sophisticated Man-in-the-Middle (MITM) attack can be mounted against NEA systems. The attacker forwards PA-TNC messages from a healthy machine through an unhealthy one so that the unhealthy machine can gain network access. Because there are easier attacks on NEA systems, like having the unhealthy machine lie about its configuration, this attack is generally only mounted against machines with an External Measurement Agent (EMA). The EMA is a separate entity, difficult to compromise, that measures and attests to the configuration of the endpoint.

To protect against NEA Asokan attacks, the Posture Broker Client on an EMA-equipped endpoint should pass the tls-unique channel binding [RFC5929] for PT-TLS's underlying TLS session to the EMA. This value can then be included in the EMA's attestation, and the Posture Validator responsible for communicating with the EMA may then confirm that the value matches the tls-unique channel binding for its end of the connection. If the values match, the posture sent by the EMA and NEA Client is from the same endpoint as the client side of the TLS connection (since the endpoint knows the tls-unique value), so no man-in-the-middle is forwarding posture. If they differ, the Asokan attack has been detected. The Posture Validator MUST fail its verification of the endpoint if the Asokan attack has been detected.

3.4. PT-TLS Message Flow

This section discusses the general flow of messages between the NEA Client's Posture Transport Client and the NEA Server's Posture Transport Server in order to perform NEA assessments using the PT-TLS protocol.

3.4.1. Assessment Triggers

Initially, the NEA Client or NEA Server will decide that an assessment is needed. What stimulates the decision to perform an assessment is outside the scope of this specification, but some examples include:

- NEA Server becoming aware of suspicious behavior on an endpoint
- NEA Server receiving new policies requiring immediate action
- NEA Client noticing a change in local security posture
- NEA Client wishing to access a protected network or resource

Because either the NEA Client or NEA Server can trigger the establishment of the TLS session and initiate the assessment, this document will use the terms "assessment initiator" and "assessment responder". This nomenclature allows either NEA component to fill either of the PT-TLS roles.

3.4.2. PT-TLS Message Exchange Phases

The PT-TLS message exchange occurs in three distinct phases:

- TLS Setup (including TLS handshake protocol)
- PT-TLS Negotiation
- PT-TLS Data Transport

The TLS Setup phase is responsible for the establishment of the TCP connection and the TLS protections for the PT-TLS messages. The TLS Setup phase starts with the establishment of a TCP connection between the Posture Transport Client and Posture Transport Server. The new connection triggers the TLS server to start the TLS handshake protocol to establish the cryptographic protections for the session. Once the TLS Setup phase has completed (after the TLS Finished messages), the TLS session MUST NOT be renegotiated. TLS session renegotiation MAY be used before the TLS Setup phase ends and the PT-TLS Negotiation phase begins. This phase also enables the establishment of the tls-unique shared secret. The tls-unique shared secret can later be used by the PA protocol to protect against some forms of man-in-the-middle attacks.

The PT-TLS Negotiation phase is only performed at the start of the first assessment on a TLS session. During this phase, the NEA Client and NEA Server discover each other's PT-TLS capabilities and establish a context that will apply to all future PT-TLS messages sent over the TLS session. The PT-TLS Negotiation phase MUST NOT be repeated after the session has entered the Data Transport phase. NEA assessment messages (PB-TNC batches) MUST NOT be sent by the NEA Client or NEA Server prior to the completion of the PT-TLS Negotiation phase to ensure that the security protections for the session are properly established and applied to the NEA assessment messages.

Finally, the Data Transport phase allows the NEA Client and NEA Server to exchange PT messages under the protection of the TLS session consistent with the capabilities established in earlier phases. The exchanged messages can be a PT-TLS protected NEA assessment as described in this specification or other vendor-defined PT-TLS exchanged messages.

3.4.2.1. TLS Setup Phase

After a new TCP connection is established between the Posture Transport Client and Posture Transport Server, a standard TLS exchange is performed to negotiate a common security context for protecting subsequent communications. As discussed in Section 3.1, the TCP connection establishment and/or the TLS handshake protocol could be initiated by either the NEA Client or NEA Server. The most common situation would be for the assessment initiator to trigger the creation of the TCP connection and TLS handshake, so an assessment could begin when no session already exists. When the NEA Server has initiated the TLS Setup, the NEA Server is acting as a TLS client and the NEA Client is the TLS server (accepting the inbound TLS session request). The expected normal case is that the NEA Client initiates this phase, so that the NEA Server is acting as the TLS server and therefore the bootstrapping of the security of the TLS session is using the NEA Server's certificate. Having the NEA Client initiate the TLS session avoids the need for the NEA Client to also possess a certificate.

During the TLS Setup phase of PT-TLS, the PT-TLS Initiator contacts the listening port of the PT-TLS Responder and performs a TLS handshake. The PT-TLS Responder MUST possess a trustworthy X.509 certificate used to authenticate to the PT-TLS Initiator and used to bootstrap the security protections of the TLS session. The PT-TLS Initiator MAY also use an X.509 certificate to authenticate to the PT-TLS Responder providing for a bidirectional authentication of the PT-TLS session. The NEA Client MUST provide certificate validation according to the rules in RFC 5280 when evaluating the server certificate. The NEA Client MAY perform certificate revocation checking on the NEA Server's certificate. This specification allows the NEA Client implementation to decide on what certificate revocation technique is to be implemented. If revocation information is not provided in a TLS handshake extension, then clients performing certificate validation will require some network access (e.g., HTTP) to be allowed during the assessment. NEA Client network access to other non-essential services might be restricted during assessments in some situations. If the client cannot access the status information, then its policy may prevent it from completing the TLS handshake.

In addition, the NEA Client MUST follow the recommendations in RFC 6125 [RFC6125] when validating the NEA Server domain name against the contents of the server certificate, taking into consideration the following restrictions:

- Any SRV-IDs and URI-IDs in the certificate are ignored.
- Use of CN-IDs in certificates is NOT RECOMMENDED.
- Wildcards MUST NOT appear in the DNS-ID or CN-ID of a certificate identifying a PT-TLS server.

Details for the reverse direction are given in Section 3.1.

Due to deployment issues with issuing and distributing certificates to a potentially large number of NEA Clients, this specification allows the NEA Client to be authenticated during the PT-TLS Negotiation phase using other more cost-effective methods, as described in Section 3.8.1. At the conclusion of a successful initial TLS Setup phase, the NEA Client and NEA Server have a protected session to exchange messages. This allows the protocol to transition to the PT-TLS Negotiation phase.

3.4.2.2. PT-TLS Negotiation Phase

Once a TLS session has been established between a Posture Transport Client and Posture Transport Server, the PT-TLS Initiator sends a Version Request message indicating its supported PT-TLS protocol version range. Next, the PT-TLS Responder sends a Version Response message, which selects a protocol version from within the range offered. The PT-TLS Responder SHOULD select the preferred version offered if supported; otherwise, the highest version that the responder is able to support from the received Version Request message will be used. If the PT-TLS Responder is unable or unwilling to support any of the versions included in the Version Request message, the responder SHOULD send a Version Not Supported error message.

If no client-side authentication occurred during the TLS Setup phase, the Posture Transport Server can authenticate the client using PT-TLS client authentication messages as described in Section 3.8. The NEA Server initiates the client authentication and indicates when the authentication is complete.

When the NEA Client receives the Simple Authentication and Security Layer (SASL) [RFC4422] Mechanisms list, the NEA Client responds with a SASL Mechanism Selection TLV indicating the method of authentication to be used. Upon selecting an appropriate SASL

mechanism, the NEA Client and Server exchange SASL-mechanism-specific messages in order to authenticate the NEA Client. When the client authentication successfully completes and no additional authentications are required (as indicated by the NEA Server sending an empty SASL Mechanisms list), the PT-TLS session transitions into the Data Transport phase, where it will remain for the duration of the session. Note that the NEA Server could choose to not authenticate the client (indicated by only sending an empty SASL Mechanisms list) or to continue performing a posture assessment even if the authentication did not complete successfully.

3.4.2.3. PT-TLS Data Transport Phase

Once a PT-TLS session is available to carry NEA assessments, PT-TLS allows either side of the connection to send the first PB-TNC batch. The PB-TNC standard prescribes whether the Posture Broker Client or Posture Broker Server starts the assessment. The assessment initiator first envelopes the PB-TNC batch in a PT-TLS message, then assigns a message identifier to the message and finally transmits it over the session. The assessment responder validates the PT-TLS message and delivers the encapsulated PB-TNC batch to its upstream component (Posture Broker Client or Server).

Most PT-TLS messages contain PB-TNC batches that house PA-TNC requests for posture information or a response containing the requested posture information. The Posture Transport Client and Posture Transport Server may also exchange messages between them, such as a PT-TLS Error message indicating that a problem occurred processing a message. During an assessment, the Posture Transport Client and Server merely encapsulate and exchange the PB-TNC batches and are unaware of the state of the assessment.

The PT-TLS protocol allows either party to send a PT-TLS message at any time, reflecting the full-duplex nature of the underlying TLS session. For example, an assessment initiator may send several PT-TLS messages prior to receiving any responses from the assessment responder. All implementations of PT-TLS MUST support full-duplex PT-TLS message exchange. However, some higher-layer NEA protocols (e.g., PB-TNC) may not be able to fully make use of the full-duplex message exchange.

3.4.3. TLS Requirements

In order to ensure that strong security is always available for deployers and to improve interoperability, this section discusses some requirements on the underlying TLS transport used by PT-TLS. Whenever TLS is used by this specification, the appropriate version (or versions) of TLS will vary over time, based on the widespread

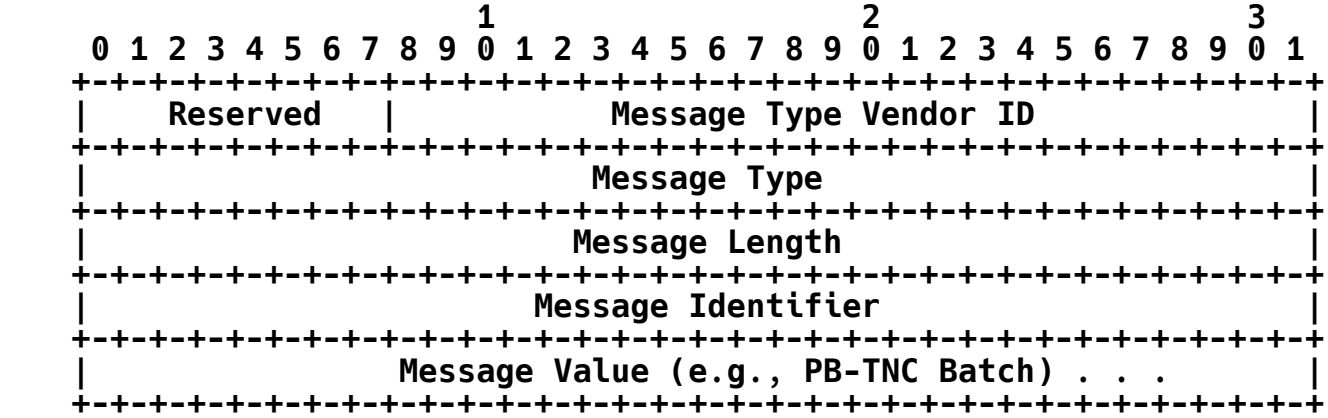
deployment and known security vulnerabilities. At the time of this writing, TLS version 1.2 [RFC5246] is the most recent version, but it has a very limited deployment base and might not be readily available for implementation. TLS version 1.0 [RFC2246] and version 1.1 [RFC4346] are the most widely deployed versions and will provide the broadest interoperability. Implementations of PT-TLS SHOULD support use of TLS 1.2.

For each TLS version supported, implementations of the PT-TLS protocol MUST at least support the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite. This cipher suite requires the server to provide a certificate that can be used during the key exchange. Implementations SHOULD NOT include support for cipher suites that do not minimally offer PT-TLS Responder (typically Posture Transport Server) authentication, such as the anonymous Diffie-Hellman cipher suites (e.g., TLS_DH_anon_WITH_AES_128_CBC_SHA). Implementations MUST support RFC 5746 [RFC5746]. Implementations MAY allow renegotiation to provide confidentiality for the client certificate. If renegotiation is allowed, implementations need to select the appropriate handshake messages as described in RFC 5929 for the tls-unique value.

3.5. PT-TLS Message Format

This section describes the format and semantics of the PT-TLS message. Every message sent over a PT-TLS session MUST start with the PT-TLS header described in this section.

The PT-TLS header format is as follows:



Reserved

Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

Message Type Vendor ID

This field indicates the owner of the namespace associated with the message type. This is accomplished by specifying the 24-bit Structure of Management Information (SMI) Private Enterprise Number [PEN] (Vendor ID) of the party who owns the message type namespace. Consistent with PA-TNC and PB-TNC, we depend on the PEN fitting in 24 bits, so if IANA were to register a wider PEN, then that PEN could not be used with NEA. IETF namespace PT-TLS Message Types MUST use zero (0) in this field. For more information about the intended use of NEA namespace identifiers, see the PA-TNC specification (RFC 5792), Sections 2.1 and 2.2.

The PT-TLS Message Type Vendor ID 0xffffff is reserved. Posture Transport Clients and Servers MUST NOT send PT-TLS messages in which the PT-TLS Message Type Vendor ID has this reserved value (0xffffff). If a Posture Transport Client or Posture Transport Server receives a message containing this reserved value (0xffffff) in the PT-TLS Message Type Vendor ID, the recipient SHOULD respond with an Invalid Parameter error code in a PT-TLS Error message.

Message Type

This field defines the type of the PT-TLS message within the scope of the specified Message Type Vendor ID that is included in the Message Value field. The specific IETF-defined values allowable in this field when the Message Type Vendor ID is the IETF SMI Private Enterprise Number value (0) are defined in Section 3.6. Recipients of a message containing a Message Type Vendor ID and a message type that is unrecognized SHOULD respond with a Type Not Supported error code in a PT-TLS Error message.

Posture Transport Clients and Posture Transport Servers MUST NOT require support for particular vendor-defined PT-TLS Message Types in order to interoperate with other PT-TLS compliant implementations (although implementations MAY permit administrators to configure them to require support for specific vendor-defined PT-TLS Message Types).

If the PT-TLS Message Type Vendor ID field has the value zero (0), then the PT-TLS Message Type field contains an IETF namespace PT-TLS Message Type, as listed in the IANA registry. IANA maintains a registry of PT-TLS Message Types. Entries in this registry are added following the IANA Specification Required policy, following the guidelines in Section 6.1. Section 3.6 of this specification defines the initial set of IETF-defined PT-TLS Message Types.

The PT-TLS Message Type 0xffffffff is reserved. Posture Transport Clients and Posture Transport Servers MUST NOT send PT-TLS messages in which the PT-TLS Message Type has this reserved value (0xffffffff). If a Posture Transport Client or Posture Transport Server receives a message in which the PT-TLS Message Type has this reserved value (0xffffffff), it SHOULD respond with an Invalid Parameter error code in a PT-TLS Error message.

Message Length

This field contains the length in octets of the entire PT-TLS message (including the entire header). Therefore, this value MUST always be at least 16. Any Posture Transport Client or Posture Transport Server that receives a message with a PT-TLS Message Length field whose value is less than 16 SHOULD respond with an Invalid Parameter PT-TLS Error Code. Similarly, if a Posture Transport Client or Posture Transport Server receives a PT-TLS message for a Message Type that has a known Message Length and the Message Length indicates a different value (greater or less than the expected value), the recipient SHOULD respond with an Invalid Parameter PT-TLS Error Code.

Message Identifier

This field contains a value that uniquely identifies the PT-TLS message on a per message sender (Posture Transport Client or Server) basis. This value is copied into the body of the PT-TLS Error message so the recipient can determine which message caused the error.

The Message Identifier MUST be a monotonically increasing counter starting at zero indicating the number of the messages the sender has transmitted over the TLS session. It is possible that a busy or long-lived session might exceed 2^32-1 messages sent, so the message sender MUST roll over to zero upon reaching the 2^32nd message, thus restarting the increasing counter. During a rollover, it is feasible that the message recipient could be confused if it keeps track of every previously received Message Identifier, so recipients MUST be able to handle rollover situations without generating errors.

Message Value

The contents of this field vary depending on the particular Message Type Vendor ID and Message Type given in the PT-TLS header for this PT-TLS message. This field most frequently contains a PB-TNC batch. The contents of this field for each of the initial IETF namespace PT-TLS Message Types are defined in this specification.

3.6. IETF Namespace PT-TLS Message Types

This section defines the NEA standard PT-TLS Message Types used to carry PT-TLS messages and PB-TNC batches between the Posture Transport Client and Posture Transport Server.

The following table summarizes the initial set of IETF-defined message type values, which are used with the PT-TLS Message Type Vendor ID field set to the IETF SMI PEN (0). Note that the IANA administers a PEN value of 0 on behalf of the IETF. These descriptions only apply to the IETF namespace.

Value (Name)	Definition
0 (Experimental)	Reserved for experimental use. This type will not offer interoperability but allows for experimentation. This message type MUST only be sent when the NEA Client and NEA Server are in the Data Transport phase and only on a restricted, experimental network. Compliant implementations MUST send an Invalid Message error code in a PT-TLS Error message if an Experimental message is received.
1 (Version Request)	Version negotiation request including the range of versions supported by the sender. This message type MUST only be sent by the TLS session initiator as the first PT-TLS message in the PT-TLS Negotiation phase. Recipients MUST send an Invalid Message error code in a PT-TLS Error message if a Version Request is received at another time.

2 (Version Response)	PT-TLS protocol version selected by the responder. This message type MUST only be sent by the PT-TLS Responder as the second message in the PT-TLS Negotiation phase. Recipients MUST send an Invalid Message error code in a PT-TLS Error message if a Version Response is received at another time.
3 (SASL Mechanisms)	Sent by the NEA Server to indicate what SASL mechanisms it is willing to use for authentication on this session. This message type MUST only be sent by the NEA Server in the PT-TLS Negotiation phase. The NEA Client MUST send an Invalid Message error code in a PT-TLS Error message if a SASL Mechanisms message is received at another time.
4 (SASL Mechanism Selection)	Sent by the NEA Client to select a SASL mechanism from the list offered by the NEA Server. This message type MUST only be sent by the NEA Client in the PT-TLS Negotiation phase. The NEA Server MUST send an Invalid Message error code in a PT-TLS Error message if a SASL Mechanism Selection is received after the PT-TLS Negotiation phase. Once a SASL mechanism has been selected, it may not change until the mechanism completes either successfully or as a failure.
5 (SASL Authentication Data)	Opaque octets exchanged between the NEA Client and NEA Server's SASL mechanisms to perform the client authentication. This message type MUST only be sent during the PT-TLS Negotiation phase. Recipients MUST send an Invalid Message error code in a PT-TLS Error message if a SASL Authentication Data message is received after the PT-TLS Negotiation phase.

6 (SASL Result)	Indicates the result code of the SASL mechanism authentication as described in Section 3.8.10. This message type MUST only be sent by the NEA Server when the NEA Client and NEA Server are in the PT-TLS Negotiation phase. The NEA Client MUST send an Invalid Message error code in a PT-TLS Error message if a SASL Result is received after the PT-TLS Negotiation phase.
7 (PB-TNC Batch)	Contains a PB-TNC batch. For more information on PB-TNC batches, see RFC 5793 (PB-TNC) Section 4. This message type MUST only be sent when the NEA Client and NEA Server are in the PT-TLS Data Transport phase. Recipients SHOULD send an Invalid Message error code in a PT-TLS Error message if a PB-TNC Batch is received outside of the Data Transport phase.
8 (PT-TLS Error)	PT-TLS Error message as described in Section 3.9. This message type may be used during any PT-TLS phase.
9-4294967294 (Unassigned)	These values are for future allocation following guidelines defined in the IANA Considerations section (see Section 6.1). Recipients of unsupported messages in the IETF namespace using a message type of 9 to 4294967294 MUST respond with a Type Not Supported PT-TLS Error Code in a PT-TLS Error message.
4294967295	Reserved

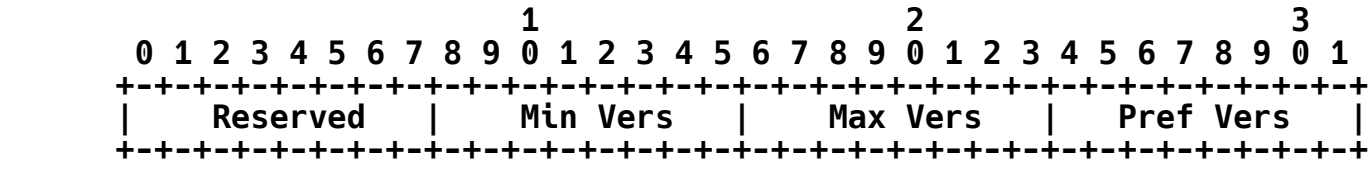
3.7. PT-TLS Version Negotiation

This section describes the message format and semantics for the PT-TLS protocol version negotiation. This exchange is used by the PT-TLS Initiator to trigger a version negotiation at the start of an assessment. The PT-TLS Initiator MUST send a Version Request message as its first PT-TLS message and MUST NOT send any other PT-TLS messages on this connection until it receives a Version Response message or an Error message. The PT-TLS Responder MUST complete the version negotiation (or cause an error) prior to sending or accepting

reception of any additional messages. After the successful completion of the version negotiation, both the Posture Transport Client and Posture Transport Server MUST only send messages compliant with the negotiated protocol version. Subsequent assessments on the same session MUST use the negotiated version number and therefore MUST NOT send additional version negotiation messages.

3.7.1. Version Request Message

This message is sent by a PT-TLS Initiator as the first PT-TLS message in a PT-TLS session. This message discloses the sender's supported versions of the PT-TLS protocol. To ensure compatibility, this message MUST always be sent using version 1 of the PT-TLS protocol. Recipients of this message MUST respond with a Version Response or with a PT-TLS Error message (Version Not Supported or Invalid Message). The following diagram shows the format of the Version Request message:



Reserved

Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

Min Vers

This field contains the minimum version of the PT-TLS protocol supported by the sender. This field MUST be set to 1 indicating support for the first version of PT-TLS. However, future versions of this specification will probably remove this requirement, so PT-TLS Responders MUST be prepared to receive other values.

Max Vers

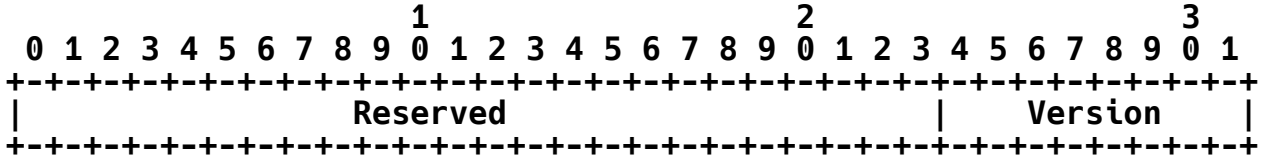
This field contains the maximum version of the PT-TLS protocol supported by the sender. This field MUST be set to 1 indicating support for the first version of PT-TLS. However, future versions of this specification will probably remove this requirement, so PT-TLS Responders MUST be prepared to receive other values.

Pref Vers

This field contains the sender's preferred version of the PT-TLS protocol. This is a hint to the recipient that the sender would like this version selected if supported. The value of this field MUST fall within the range of Min Vers to Max Vers. This field MUST be set to 1 indicating support for the first version of PT-TLS. However, future versions of this specification will probably remove this requirement, so PT-TLS Responders MUST be prepared to receive other values.

3.7.2. Version Response Message

This message is sent in response to receiving a Version Request message at the start of a new assessment session. If a recipient receives a Version Request after a successful version negotiation has occurred on the session, the recipient MUST send an Invalid Message error code in a PT-TLS Error message and have TLS close the session. This message MUST be sent using the syntax, semantics, and requirements of the protocol version specified in this message.



Reserved

Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

Version

This field contains the version selected by the sender of this message. The version selected MUST be within the Min Vers to Max Vers inclusive range sent in the Version Request message. If a PT-TLS Initiator receives a message with an invalid Version selected, the PT-TLS Initiator MUST respond with a Version Not Supported PT-TLS Error message.

3.8. Client Authentication Using SASL

This section includes a description of the message format and semantics necessary to perform client authentication (authentication of the NEA Client) over PT-TLS. Client authentication could be necessary if the NEA Server requires such an authentication and it was not performed during the TLS handshake. The general model used

for performing an authentication of the client using PT-TLS messages is to integrate the Simple Authentication and Security Layer (SASL) [RFC4422] framework. SASL provides a number of standards-based authentication mechanisms capable of authenticating the NEA Client using a variety of base technologies.

Client authentication could occur during the TLS handshake using TLS-defined authentication techniques. Because this client authentication is optional, the NEA Server's policy might require the client to be authenticated by PT-TLS before performing the assessment. Similarly, the NEA Server may require a PT-TLS authentication even if the NEA Client was authenticated during the TLS handshake (e.g., to allow a user authentication after a system-level authentication occurred during the TLS handshake). The decision of whether a SASL client authentication is to occur is left to the NEA Server's policy.

As discussed in Section 3.1.1, it is possible that the NEA Server may initiate the TLS session to the NEA Client, thus causing it to fill the role of TLS client during the TLS handshake. Because the NEA Server is required to possess an X.509 certificate for those times when it is acting as the TLS server (normal case), PT-TLS requires that the NEA Server MUST use its X.509 certificate for TLS client authentication during the TLS handshake to authenticate itself even when it is acting as the TLS client. In this case, the NEA Client and NEA Server will authenticate using certificates during the TLS handshake, so the PT-TLS SASL client authentication might not be required unless NEA Server policy required an additional authentication of the NEA Client. Therefore, the normal usage for the SASL messages is when the NEA Client acted as the TLS client and did not authenticate during the TLS handshake.

3.8.1. SASL Client Authentication Requirements

Implementations compliant with the PT-TLS specification MUST implement the PT-TLS client authentication messages described in this section. These PT-TLS client authentication messages are capable of carrying a variety of SASL authentication mechanisms' exchanges. In order to ensure interoperability, all PT-TLS implementations compliant with this specification MUST at least support the PLAIN SASL mechanism [RFC4616]. Similarly, implementations MUST provide the EXTERNAL SASL mechanism if both parties are authenticated during the TLS establishment. In order to be able to take advantage of other strong, widely deployed authentication technologies such as Kerberos and support for channel bindings, implementations MAY

include support for GS2 (the second Generic Security Service Application Program Interface (GSS-API) bridge for SASL) [RFC5801]. GS2 includes negotiable support for channel binding for use with SASL (see Section 5 of RFC 5801).

Implementations MUST also support the case where no client authentication is required.

3.8.2. SASL in PT-TLS Overview

Mechanism negotiation is initiated by the NEA Server sending the SASL Mechanisms TLV to the NEA Client to indicate the zero or more SASL mechanisms that the NEA Server's policy is willing to use with the NEA Client. The NEA Client selects one SASL mechanism from the list and sends a SASL Mechanism Selection TLV completing the negotiation. Subsequent challenges and responses are carried within the SASL Authentication Data TLV carrying the authentication data for the selected mechanism. The authentication outcome is communicated in a SASL Result TLV containing a status code. If additional authentications are required, the NEA Server could trigger the next authentication by sending another SASL Mechanisms TLV after sending the SASL Result TLV for the current authentication mechanism.

3.8.3. SASL Authentication Flow

The SASL client authentication starts when the NEA Server enters the PT-TLS Negotiation phase and its policy indicates that an authentication of the NEA Client is necessary, such as if it was not performed during the TLS handshake protocol. The NEA Server is responsible for triggering the client authentication by sending the SASL Mechanisms TLV to the NEA Client listing the set of SASL mechanisms the server is willing to use based upon its policy.

The NEA Client selects a SASL mechanism from the list proposed by the NEA Server or sends a PT-TLS Invalid Message error code indicating that it is unable or unwilling to perform any of the mechanisms that were offered. If the NEA Server receives a SASL Mechanism Selection TLV that contains an unacceptable SASL mechanism, the NEA Server would respond with a SASL Mechanism Error in a PT-TLS Error TLV.

In situations where the NEA Server does not require a client authentication, the NEA Server MUST send a SASL Mechanisms TLV with no mechanisms included (only the PT-TLS header) indicating that the connection should transition to the PT-TLS Data Transport phase. The same mechanism is employed to indicate that a SASL authentication already performed in this session is adequate to permit transition to the PT-TLS Data Transport phase. So the NEA Server MUST always send

a SASL Mechanisms TLV with no mechanisms as the last message in the PT-TLS Negotiation phase, and the NEA Client MUST NOT transition to the PT-TLS Data Transport phase until it receives such a message.

If the NEA Server receives a NEA assessment message before the completion of the client authentication, the NEA Server MUST send an Authentication Required PT-TLS Error message indicating to the NEA Client that an authentication exchange is required prior to entering the PT-TLS Data Transport phase.

3.8.4. Aborting SASL Authentication

The NEA Server may abort the authentication exchange by sending the SASL Result TLV with a status code of Abort. The NEA Client may abort the authentication exchange by sending a PT-TLS Error message with an Error Code of SASL Mechanism Error.

3.8.5. Linkages to SASL Framework

3.8.5.1. SASL Service Name

The service name for PT-TLS is "nea-pt-tls".

3.8.5.2. SASL Authorization Identity

The PT-TLS protocol does not make use of a SASL authorization identity string as described in RFC 4422.

3.8.5.3. SASL Security Layer

The NEA PT-TLS protocol always runs under the protection of TLS. SASL security layers are not used and thus MUST be negotiated off during SASL authentication.

3.8.5.4. Multiple Authentications

Only one SASL mechanism authentication may be in progress at any one time. Once a SASL mechanism completes (successfully or unsuccessfully), the NEA Server MAY trigger an additional authentication by sending a SASL Mechanisms TLV.

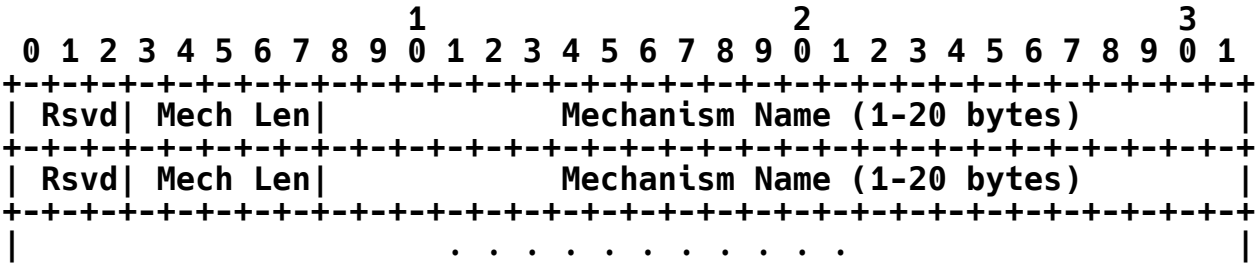
3.8.6. SASL Channel Bindings

SASL channel bindings are used to bind the SASL authentication to the outer TLS tunnel to ensure that the authenticating endpoints are the same as the TLS endpoints. For SASL mechanisms that support channel bindings, the tls-unique value defined in RFC 5929 is carried by the SASL mechanism. For most mechanisms, this means including the

tls-unique value with the appropriate prefix defined in RFC 5929 in the application data portion of the SASL Mechanism channel-binding data. If the validation of the channel binding fails, then the connection MUST be aborted.

3.8.7. SASL Mechanisms

This TLV is sent by the NEA Server to indicate the list of SASL mechanisms that it is willing and able to use to authenticate the NEA Client. Each mechanism name consists of a length followed by a name. The total length of the list is determined by the TLV Length field.



Rsvd (Reserved)

Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

Mech Len (Mechanism Name Length)

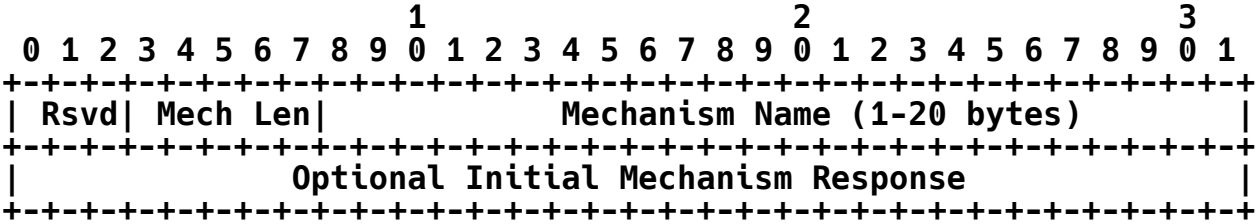
The length of the Mechanism Name field in octets.

Mechanism Name

SASL mechanism name adhering to the rules defined in RFC 4422 with no padding.

3.8.8. SASL Mechanism Selection

This TLV is sent by the NEA Client in order to select a SASL mechanism for use on this session.



Rsvd (Reserved)

Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

Mech Len (Mechanism Name Length)

The length of the Mechanism Name field in octets.

Mechanism Name

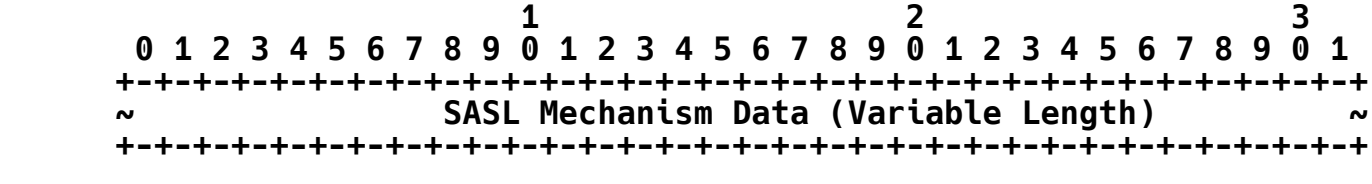
SASL mechanism name adhering to the rules defined in RFC 4422.

Optional Initial Mechanism Response

Initial set of authentication information required from the NEA Client to kick-start the authentication. This data is optional and if not provided would be solicited by the NEA Server in the first SASL Authentication Data TLV request.

3.8.9. SASL Authentication Data

This TLV carries an opaque (to PT-TLS) blob of octets being exchanged between the NEA Client and the NEA Server. This TLV facilitates their communications without interpreting any of the bytes. The SASL Authentication Data TLV MUST NOT be sent until a SASL mechanism has been established for a session. The SASL Authentication Data TLV associated with the current authentication mechanism MUST NOT be sent after a SASL Result is sent with a Result Code of Success. Additional SASL Authentication Data TLVs would be sent if the PT-TLS Initiator and Responder desire a subsequent SASL authentication to occur but only after another SASL mechanism selection exchange occurs.

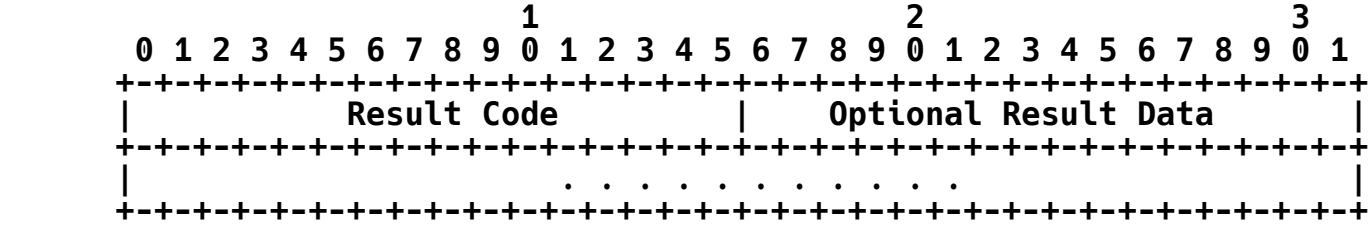


SASL Mechanism Data

Opaque, variable-length set of bytes exchanged between the PT-TLS Initiator's SASL mechanism and its peer PT-TLS Responder's SASL mechanism. These bytes MUST NOT be interpreted by the PT-TLS layer.

3.8.10. SASL Result

This TLV is sent by the NEA Server at the conclusion of the SASL exchange to indicate the authentication result. Upon reception of a SASL Result TLV indicating an Abort, the NEA Client MUST terminate the current authentication conversation. The recipient may retry the authentication in the event of an authentication failure. Similarly, the NEA Server may request that additional SASL authentication(s) be performed after the completion of a SASL mechanism by sending another SASL Mechanisms TLV including any mechanisms dictated by its policy.



Result Code

This field contains the result of the SASL authentication exchange.

Value (Name)	Definition
0 (Success)	SASL authentication was successful, and identity was confirmed.
1 (Failure)	SASL authentication failed. This might be caused by the client providing an invalid user identity and/or credential pair. Note that this is not the same as a mechanism's failure to process the authentication as reported by the Mechanism Failure code.
2 (Abort)	SASL authentication exchange was aborted by the sender.
3 (Mechanism Failure)	SASL "mechanism failure" during the processing of the client's authentication (e.g., not related to the user's input). For example, this could occur if the mechanism is unable to allocate memory (e.g., malloc) that is needed to process a received SASL mechanism message.

Optional Result Data

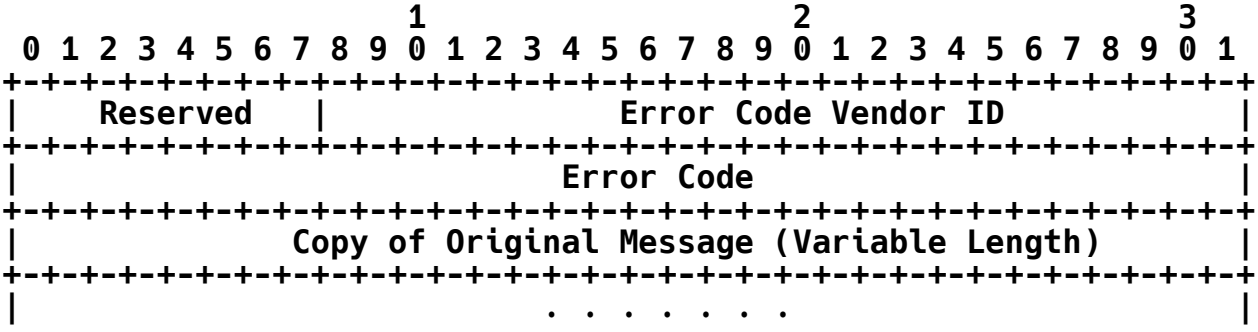
This field contains a variable-length set of additional data for a successful result. This field MUST be zero length unless the NEA Server is returning a Result Code of Success and has more data to return. For more information on the additional data with success in SASL, see RFC 4422.

3.9. Error Message

This section describes the format and contents of the PT-TLS Error message sent by the NEA Client or NEA Server when it detects a PT-TLS-level protocol error. Each error message contains an error code indicating the error that occurred, followed by a copy of the message that caused the error.

When a PT-TLS error is received, the recipient MUST NOT respond with a PT-TLS error, because this could result in an infinite loop of error messages being sent. Instead, the recipient MAY log the error, modify its behavior to avoid future errors, ignore the error, terminate the assessment, or take other action as appropriate (as long as it is consistent with the requirements of this specification).

The Message Value portion of a PT-TLS Error message contains the following information:



Reserved

Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

Error Code Vendor ID

This field contains the IANA-assigned SMI Private Enterprise Number for the vendor whose Error Code namespace is being used in the message. For IETF namespace Error Code values, this field MUST be set to zero (0). For other vendor-defined Error Code namespaces, this field MUST be set to the SMI Private Enterprise Number of the vendor.

Error Code

This field contains the error code. This error code exists within the scope of Error Code Vendor ID in this message. Posture Transport Clients and Posture Transport Servers MUST NOT require support for particular vendor-specific PT-TLS Error Codes in order to interoperate with other PT-TLS-compliant implementations (although implementations MAY permit administrators to configure them to require support for specific PT-TLS Error Codes).

When the Error Code Vendor ID is set to the IETF Private Enterprise Number, the following table lists the initial supported IETF-defined numeric error codes:

Value (Name)	Definition
0 (Reserved)	Reserved value indicates that the PT-TLS Error message SHOULD be ignored by all recipients. This MAY be used for debugging purposes to allow a sender to see a copy of the message that was received.
1 (Malformed Message)	PT-TLS message unrecognized or unsupported. This error code SHOULD be sent when the basic message content sanity test fails. The sender of this error code MUST consider it a fatal error and abort the assessment.
2 (Version Not Supported)	This error SHOULD be sent when a PT-TLS Responder receives a PT-TLS Version Request message containing a range of version numbers that doesn't include any version numbers that the recipient is willing and able to support on the session. All PT-TLS messages carrying the Version Not Supported error code MUST use a version number of 1. All

	parties that receive or send PT-TLS messages MUST be able to properly process an error message that meets this description, even if they cannot process any other aspect of PT-TLS version 1. The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this PT-TLS message.
3 (Type Not Supported)	PT-TLS Message Type unknown or not supported. When a recipient receives a PT-TLS Message Type that it does not support, it MUST send back this error, ignore the message, and proceed. For example, this could occur if the sender used a Vendor ID for the Message Type that is not supported by the recipient. This error message does not indicate that a fatal error has occurred, so the assessment is allowed to continue.
4 (Invalid Message)	PT-TLS message received was invalid based on the protocol state. For example, this error would be sent if a recipient receives a message associated with the PT-TLS Negotiation Phase (such as Version messages) after the protocol has reached the PT-TLS Data Transport Phase. The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this PT-TLS message.
5 (SASL Mechanism Error)	A fatal error occurred while trying to perform the client authentication. For example, the NEA Client is unable to support any of the offered SASL mechanisms. The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this PT-TLS message.

6 (Invalid Parameter)	The PT-TLS Error message sender has received a message with an invalid or unsupported value in the PT-TLS header. This could occur if the NEA Client receives a PT-TLS message from the NEA Server with a Message Length of zero (see Section 3.5 for details). The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this PT-TLS message.

Copy of Original Message

This variable-length value MUST contain a copy (up to 1024 bytes) of the original PT-TLS message that caused the error. If the original message is longer than 1024 bytes, only the initial 1024 bytes will be included in this field. This field is included so the error recipient can determine which message sent caused the error. In particular, the recipient can use the Message Identifier field from the Copy of Original Message data to determine which message caused the error.

4. Security Considerations

This section discusses the major threats potentially faced by each binding of the PT protocol and countermeasures provided by the PT-TLS protocol.

4.1. Trust Relationships

In order to understand where security countermeasures are necessary, this section starts with a discussion of where the NEA architecture envisions some trust relationships between the processing elements of the PT-TLS protocol. Implementations or deployments where these trust relationships are not present would need to provide additional countermeasures to ensure the proper operation and security of PT-TLS (which relies upon these relationships to be trustworthy). The following subsections discuss the trust properties associated with each portion of the NEA reference model directly involved with the processing of the PT-TLS protocol.

4.1.1. Posture Transport Client

The Posture Transport Client is trusted by the Posture Broker Client to:

- Not observe, fabricate, or alter the contents of the PB-TNC batches received from the network
- Not observe, fabricate, or alter the PB-TNC batches passed down from the Posture Broker Client for transmission on the network
- Transmit on the network any PB-TNC batches passed down from the Posture Broker Client
- Deliver properly security protected messages received from the network that are destined for the Posture Broker Client
- Provide configured security protections (e.g., authentication, integrity, and confidentiality) for the Posture Broker Client's PB-TNC batches sent on the network
- Expose the authenticated identity of the Posture Transport Server only to the PB-TNC layer within the NEA Client
- Verify the security protections placed upon messages received from the network to ensure that the messages are authentic and protected from attacks on the network
- Provide a secure, reliable, "in-order delivery", full-duplex transport for the Posture Broker Client's messages

The Posture Transport Client is trusted by the Posture Transport Server to:

- Not send malicious traffic intending to harm (e.g., denial of service) the Posture Transport Server
- Not send malformed messages (e.g., messages lacking a PT-TLS header)
- Not send invalid or incorrect responses to messages (e.g., errors when no error is warranted)
- Not ignore or drop messages when such an action would cause issues for the protocol processing (e.g., dropping PT-TLS SASL Authentication Data messages)

- Verify the security protections placed upon messages received from the network to ensure that the messages are authentic and protected from attacks on the network

4.1.2. Posture Transport Server

The Posture Transport Server is trusted by the Posture Broker Server to:

- Not observe, fabricate, or alter the contents of the PB-TNC batches received from the network
- Not observe, fabricate, or alter the PB-TNC batches passed down from the Posture Broker Server for transmission on the network
- Transmit on the network any PB-TNC batches passed down from the Posture Broker Server
- Deliver properly security protected messages received from the network that are destined for the Posture Broker Server
- Provide configured security protections (e.g., authentication, integrity, and confidentiality) for the Posture Broker Server's messages sent on the network
- Expose the authenticated identity of the Posture Transport Client only to the PB-TNC layer within the NEA Server
- Verify the security protections placed upon messages received from the network to ensure that the messages are authentic and protected from attacks on the network
- Provide a secure, reliable, "in-order delivery", full-duplex transport for the Posture Broker Server's messages

The Posture Transport Server is trusted by the Posture Transport Client to:

- Not send malicious traffic intending to harm (e.g., denial of service) the Posture Transport Client
- Not send malformed messages (e.g., messages lacking a PT-TLS header)
- Not send invalid or incorrect responses to messages (e.g., errors when no error is warranted)

- Not ignore or drop messages when such an action would cause issues for the protocol processing (e.g., dropping PT-TLS SASL Result messages)

- Verify the security protections placed upon messages received from the network to ensure that the messages are authentic and protected from attacks on the network

4.2. Security Threats and Countermeasures

Beyond the trust relationships assumed in Section 4.1, the PT-TLS protocol faces a number of potential security attacks that could require security countermeasures.

Generally, the PT-TLS protocol is responsible for offering strong security protections for all of the NEA protocols, so any threats to its ability to protect NEA protocol messages could be very damaging to deployments. Once the message is delivered to the Posture Broker Client or Posture Broker Server, the posture brokers are trusted to properly and safely process the messages.

4.2.1. Message Theft

When PT-TLS messages are sent over unprotected network links or spanning local software stacks that are not trusted, the contents of the messages may be subject to information theft by an intermediary party. This theft could result in information being recorded for future use or analysis by the adversary. Messages observed by eavesdroppers could contain information that exposes potential weaknesses in the security of the endpoint, or system fingerprinting information; this information would make it easier for the attacker to employ attacks more likely to be successful against the endpoint. The eavesdropper might also learn information about the endpoint or network policies that either singularly or collectively is considered sensitive information. For example, if PT-TLS does not provide confidentiality protection, an adversary could observe the PA-TNC attributes included in the PT-TLS message and determine that the endpoint is lacking patches or that particular sub-networks have more lenient policies.

In order to protect against NEA assessment message theft, the PT-TLS protocol provides strong cryptographic authentication, integrity, and confidentiality protection. Deployers are strongly encouraged to employ "best practice of the day" TLS ciphers to ensure that the information remains safe despite advances in technology and discovered cipher weaknesses. The use of bidirectional authentication of the assessment transport session ensures that only properly authenticated and authorized parties may be involved in an

assessment dialog. The PT-TLS protocol also provides strong cryptography for all of the PB-TNC and PA-TNC protocol messages traveling over the network, allowing the message contents to be hidden from potential theft by the adversary even if the attacker is able to observe the encrypted PT-TLS session.

4.2.2. Message Fabrication

Attackers on the network or present within the NEA system could introduce fabricated PT-TLS messages intending to trick or create a denial of service against aspects of an assessment. For example, an adversary could attempt to insert into the message exchange fake PT-TLS Error Codes in order to disrupt communications.

The PT-TLS protocol provides strong security protections for the complete message exchange over the network. These security protections prevent an intermediary from being able to insert fake messages into the assessment. In particular, TLS's use of hashing algorithms provides strong integrity protections that allow for detection of any changes in the content of the message stream. Additionally, adversaries are unable to observe the PT-TLS protocol exchanges because they are encrypted by the TLS ciphers and so would have difficulty determining where to insert the falsified message, since the attacker is unable to determine where the message boundaries exist. Even if a successful message insertion did occur, the recipient would be able to detect it due to failure of the TLS cipher suite's integrity check.

4.2.3. Message Modification

This attack could allow an active attacker capable of intercepting a message to modify a PT-TLS message or transported PA-TNC attribute to a desired value to make it easier to compromise an endpoint. Without the ability for message recipients to detect whether a received message contains the same content as what was originally sent, active attackers can stealthily modify the attribute exchange.

The PT-TLS protocol leverages the TLS protocol to provide strong authentication and integrity protections as a countermeasure to this threat. The bidirectional authentication prevents the attacker from acting as an active man-in-the-middle to the protocol that could be used to modify the message exchange. The strong integrity protection (e.g., hashing) offered by TLS allows PT-TLS message recipients to detect message alterations by other types of network-based adversaries.

4.2.4. Denial of Service

A variety of types of denial-of-service attacks are possible against the PT-TLS protocol if the message exchanges are left unprotected while traveling over the network. The Posture Transport Client and Posture Transport Server are trusted not to participate in the denial of service of the assessment session, leaving the threats to come from the network.

The PT-TLS protocol provides bidirectional authentication capabilities in order to prevent a man-in-the-middle on the network from becoming an undetected active proxy of PT-TLS messages. Because the PT-TLS protocol runs after the TLS handshake and thus cipher establishment/use, all of the PT-TLS messages are protected from undetected modification that could create a denial-of-service situation. However, it is possible for an adversary to alter the message flows, causing each message to be rejected by the recipient because it fails the integrity checking.

The PT-TLS protocol operates as an application protocol on top of TLS and thus TCP/IP protocols, so is subject to denial-of-service attacks against the TLS, TCP, and IP protocols.

4.2.5. NEA Asokan Attacks

As described in Section 3.3 and in [RFC6813], a sophisticated MITM attack can be mounted against NEA systems. The attacker forwards PA-TNC messages from a healthy machine through an unhealthy one so that the unhealthy machine can gain network access. Section 3.3 and [RFC6813] provide a detailed description of this attack and of the countermeasures that can be employed against it.

Because lying endpoint attacks are much easier than Asokan attacks and the only known effective countermeasure against lying endpoint attacks is the use of an External Measurement Agent (EMA), countermeasures against an Asokan attack are not necessary unless an EMA is in use. However, PT-TLS implementers may not know whether an EMA will be used with their implementation. Therefore, PT-TLS implementers SHOULD support the Asokan attack countermeasures described in Section 3.3 by providing the value of the tls-unique channel binding to higher layers in the NEA reference model: Posture Broker Clients, Posture Broker Servers, Posture Collectors, and Posture Validators.

The Asokan attack can also apply to authentication mechanisms carried within TLS. SASL mechanisms providing channel bindings use the tls-unique channel-binding data as defined in Section 3.3 to protect against the attack.

4.2.6. Trust Anchors

The TLS protocol bases its trust decision about the signer of the certificates received during the TLS authentication on using a set of trust anchor certificates. It is essential that these trust anchor certificates are integrity protected from unauthorized modification. Many common software components (e.g., browsers, operating systems, security protocols) include a set of trust anchor certificates that are relevant to their operation. The PT-TLS SHOULD use a PT-TLS-specific set of trust anchor certificates in order to limit what Certificate Authorities are authorized to issue certificates for use with NEA.

5. Privacy Considerations

The role of PT-TLS is to act as a secure transport for PB-TNC and other higher-layer protocols. As such, PT-TLS does not directly utilize personally identifiable information (PII) except when client authentication is enabled. When client authentication is being used, the NEA Client will be asked to use SASL, which may disclose a local identifier (e.g., username) associated with the endpoint and an authenticator (e.g., password) to authenticate that identity. Because the identity and authenticator are potentially privacy-sensitive information, the NEA Client MUST offer a mechanism to restrict which NEA Servers will be sent this information. Similarly, the NEA Client SHOULD provide an indication to the person being identified that a request for their identity has been made in case they choose to opt out of the authentication to remain anonymous unless no user interface is available. PT-TLS provides cryptographic peer authentication, message integrity, and data confidentiality protections to higher-layer NEA protocols that may exchange data potentially including PII. These security services can be used to protect any PII involved in an assessment from passive and active attackers on the network. Endpoints sending potentially privacy-sensitive information SHOULD ensure that the PT-TLS security protections (TLS cipher suites) negotiated for an assessment of the endpoint are adequate to avoid interception and off-line attacks of any long-term privacy-sensitive information unless other network protections are already present.

6. IANA Considerations

Per this specification, two new IANA registries have been created and a TCP port number has been assigned. IANA has permanently reserved the early allocated TCP port number 271 for use with the PT-TLS protocol.

This section defines the contents of two new IANA registries, PT-TLS Message Types and PT-TLS Error Codes, and explains how these registries work.

Each of the registries defined in this document support IETF-defined values and vendor-defined values. To explain this phenomenon, we will use the PT-TLS Message Type as an example, but the other registry works the same way.

Whenever a PT-TLS Message Type appears on a network, it is always accompanied by an SMI Private Enterprise Number (PEN), also known as a vendor ID. If this vendor ID is zero, the accompanying PT-TLS Message Type is an IETF namespace value listed in the IANA registry for PT-TLS Message Types, and its meaning is defined in the specification listed for that PT-TLS Message Type in that registry. If the vendor ID is not zero, the meaning of the PT-TLS Message Type is defined by the vendor identified by the vendor ID (as listed in the IANA registry for SMI PENs). The identified vendor is encouraged but not required to register with IANA some or all of the PT-TLS Message Types used with their vendor ID and publish a specification for each of these values.

6.1. Designated Expert Guidelines

For each of the IANA registries defined by this specification, new values are added to the registry by following the IANA Specification Required policy [RFC5226].

This section provides guidance to designated experts so that they may make decisions using a philosophy appropriate for these registries.

The registries defined in this document have plenty of values. In most cases, the IETF has approximately 2^32 values available for it to define, and each vendor has the same number of values for its use. Because there are so many values available, designated experts should not be terribly concerned about exhausting the set of values.

Instead, designated experts should focus on the following requirements. All values in these IANA registries are required to be documented in a specification that is permanently and publicly available. IETF namespace values must also be useful not harmful to the Internet, and defined in a manner that is clear and likely to ensure interoperability.

Designated experts should encourage vendors to avoid defining similar but incompatible values and instead agree on a single IETF-reviewed approach and value. However, it is beneficial to document existing practice.

There are several ways to ensure that a specification is permanently and publicly available. It may be published as an RFC. Alternatively, it may be published in another manner that makes it freely available to anyone. However, in this latter case, the vendor will need to supply a copy to the IANA and authorize the IANA to archive this copy and make it freely available to all if at some point the document becomes no longer freely available to all through other channels.

The following two sections provide guidance to the IANA in creating and managing the new IANA registries defined by this specification.

6.2. Registry for PT-TLS Message Types

The name for this registry is "PT-TLS Message Types". Each entry in this registry should include a human-readable name, an SMI Private Enterprise Number, a decimal integer value between 0 and 4294967294, and a reference to the specification where the contents of this message type are defined. This specification must define the meaning of the PT-TLS Message Type and the format and semantics of the PT-TLS Message Value field that include the designated Private Enterprise Number in the PT-TLS Message Type Vendor ID field and the designated numeric value in the PT-TLS Message Type field.

The following entries for this registry are defined in this document. Additional entries to this registry are added by following the IANA Specification Required policy, consistent with the guidelines in Section 6.1.

PEN	Value	Name	Reference
0	0	Experimental	RFC 6876
0	1	Version Request	RFC 6876
0	2	Version Response	RFC 6876
0	3	SASL Mechanisms	RFC 6876
0	4	SASL Mechanism Selection	RFC 6876
0	5	SASL Authentication Data	RFC 6876
0	6	SASL Result	RFC 6876
0	7	PB-TNC Batch	RFC 6876
0	8	PT-TLS Error	RFC 6876
0	9-4294967294	Unassigned	
0	4294967295	Reserved	RFC 6876

The PEN 0 (IETF) PT-TLS Message Type values between 9 and 4294967294 inclusive are allocated for future assignment by the IANA.

6.3. Registry for PT-TLS Error Codes

The name for this registry is "PT-TLS Error Codes". Each entry in this registry should include a human-readable name, an SMI Private Enterprise Number, a decimal integer value between 0 and 4294967295, and a reference to the specification where this error code is defined. This specification must define the meaning of this error code, a PT-TLS Message Type of PT-TLS Error, the designated Private Enterprise Number in the PT-TLS Error Code Vendor ID field, and the designated numeric value in the PT-TLS Error Code field.

The following entries for this registry are defined in this document. Additional entries to this registry are added following the IANA Specification Required policy, consistent with the guidelines in Section 6.1.

PEN	Value	Name	Reference
0	0	Reserved	RFC 6876
0	1	Malformed Message	RFC 6876
0	2	Version Not Supported	RFC 6876
0	3	Type Not Supported	RFC 6876
0	4	Invalid Message	RFC 6876
0	5	SASL Mechanism Error	RFC 6876
0	6	Invalid Parameter	RFC 6876
0	7-4294967295	Unassigned	

The PEN 0 (IETF) PT-TLS Error Codes between 7 and 4294967295 inclusive are allocated for future assignment by the IANA.

7. Acknowledgments

Thanks to the Trusted Computing Group for contributing the initial text upon which this document was based [IFT-TLS].

The authors of this document would also like to acknowledge the following people who have contributed to or provided substantial input on the preparation of this document or predecessors to it: Syam Appala, Stuart Bailey, Lauren Giroux, Steve Hanna, Josh Howlett, Scott Kelly, Carolin Latze, Sung Lee, Lisa Lorenzin, Alexey Melnikov, Ravi Sahita, Subbu Srinivasan, Susan Thomson, and Mark Townsend.

8. References

8.1. Normative References

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

[RFC4422] Melnikov, A., Ed., and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4616] Zeilenga, K., Ed., "The PLAIN Simple Authentication and Security Layer (SASL) Mechanism", RFC 4616, August 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, 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.

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

[RFC5792] Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5792, March 2010.

[RFC5793] Sahita, R., Hanna, S., Hurst, R., and K. Narayan, "PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5793, March 2010.

[RFC5929] Altman, J., Williams, N., and L. Zhu, "Channel Bindings for TLS", RFC 5929, July 2010.

[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, March 2011.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, February 2012.

8.2. Informative References

[IFT-TLS] Trusted Computing Group, "TNC IF-T: Binding to TLS", <http://www.trustedcomputinggroup.org/>, May 2009.

[PEN] IANA Private Enterprise Numbers (PEN) registry, <http://www.iana.org/assignments/enterprise-numbers>.

[PT-EAP] Cam-Winget, N. and P. Sangster, "PT-EAP: Posture Transport (PT) Protocol For EAP Tunnel Methods", Work in Progress, January 2013.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", RFC 5209, June 2008.

[RFC5801] Josefsson, S. and N. Williams, "Using Generic Security Service Application Program Interface (GSS-API) Mechanisms in Simple Authentication and Security Layer (SASL): The GS2 Mechanism Family", RFC 5801, July 2010.

[RFC6813] Salowey, J. and S. Hanna, "The Network Endpoint Assessment (NEA) Asokan Attack Analysis", RFC 6813, December 2012.

Authors' Addresses

Paul Sangster
Symantec Corporation
6825 Citrine Dr.
Carlsbad, CA 92009

EMail: paul_sangster@symantec.com

Nancy Cam-Winget
Cisco Systems
80 West Tasman Drive
San Jose, CA 95134
US

EMail: ncamwing@cisco.com

Joseph Salowey
Cisco Systems
2901 Third Avenue
Seattle, WA 98121
US

EMail: jsalowey@cisco.com