Network Working Group   N. Cam-Winget
Internet-Draft   J. Salowey
Intended status: Standards Track   H. Zhou
Expires: September 8, 2011   Cisco Systems
                                                        March 7, 2011

           Transport Layer Security (TLS) Based Transports for Network Endpoint
                     Assessment (NEA) Protocol Exchanges
                        draft-cam-winget-eap-tlv-03

Abstract

   This document describes how Network Endpoint Assessment (NEA) data
   can be carried under the protection of a Transport Layer Security
   (TLS) secured tunnel. This document defines NEA transports for TLS-
   based Extensible Authentication Protocol (EAP) tunnel methods and for
   TLS used over TCP.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

   NEA has standardized a transport agnostic Posture Broker (PB)
   protocol defined in [RFC5793] to effect a network endpoint assessment
   between a Posture Broker Client and a Posture Broker Server. These
   PB messages can be transported inside the already defined Type-
   Length-Value containers in existing TLS-based tunne EAP methods such
   as PEAP, EAP-FAST and TTLS. Similarly, this document also defines a
   TCP based transport, PT-TCP, that uses TLVs encapsulated within TLS.

2. Specification Requirements

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC2119] .

3. Protocol Layering Model

   When using EAP as the transport, the PB messages can be encapsulated
   in the TLVs defined by the tunnel EAP methods. For TLS a new TLV
   container is defined to facilitate the PB transport over TCP. The
   following diagram demonstrates the relationship between protocols
   when an EAP tunnel method is used:


```
     +------------------------------------------------------------+
     |         TLV Encapsulation of PB-PA message                 |
     |------------------------------------------------------------|
     |                       TLS                                  |
     |------------------------------------------------------------|
     |         EAP tunnel based method                  |         |
     |------------------------------------------------------------|
     |                       EAP                                  |
     |------------------------------------------------------------|
     |     Carrier Protocol (EAPOL, RADIUS, Diameter, etc.)       |
     +------------------------------------------------------------+
```
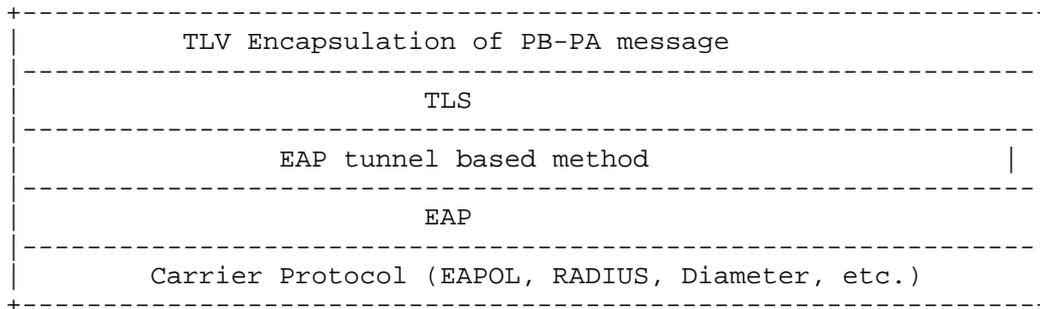

                EAP based Protocol Layering Model

   The following diagram demonstrates the protocol relationship of PB
   when PT-TCP is used:

```
+---------------------------------------------------------------+
|            TLV Encapsulation of PB-PA message                 |
|---------------------------------------------------------------|
|                          TLS                                  |
+---------------------------------------------------------------+
|                          TCP                                  |
+---------------------------------------------------------------+
```
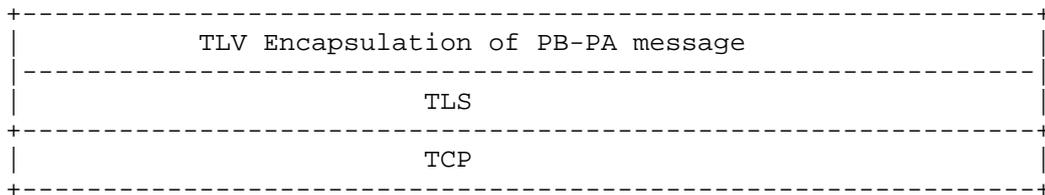
                  PT-TCP based Protocol Layering Model

4. Protocol Flows

   There are two distinct phases in TLS based transport operation:

   1. TLS Setup Phase: are the messages used to establish TLS channel
       protection for the posture messages. The TLS Setup Phase begins
       with either the Posture Transport Client or Posture Transport
       Server initiating the TLS Handshake protocol to establish the
       protected TLS channel.

   2. Data Transport Phase: are the messages that are protected by the
       TLS Record encapsulation. This phase is usually broken up into
       an optional entity authentication phase followed by the exchange
       of TLVs carrying NEA data.

4.1. PT-TCP Protocol Flow

   This section describes the general flow of messages between the NEA
   Posture Transport Client and the NEA Posture Transport Server.

4.1.1. Initiating a PT-TCP session

   With the use of TLS as the transport, it is possible for either the
   Posture Transport Client or the Posture Transport Server to initiate
   a PT-TCP session.

4.1.2. TCP Port Usage

   IANA is requested to allocate a TCP Port number for the use of PT-TCP
   so that both the Posture Transport Client and Posture Transport
   Server can communicate on a known allocated port.

4.1.3. TLS Setup Phase

   Typically, it is the NEA Client (e.g. the Posture Transport Client)
   that initiates the TLS Setup Phase. However, either party, e.g. the

Posture Transport Client or the Posture Transport Server may
establish a TCP connection and initiate the TLS Handshake protocol.
Furthermore, the TLS Handshake protocol is also used to establish the
cryptographic protections used to secure the data carried within TLS
Records.

In typical deployments, it is expected for the initiator of a NEA
exchange to initiate the TLS Setup. However, this specification
allows for multiple NEA data transactions and as such, each
transaction may originate from either the NEA client or the NEA
server. Furthermore, through the use of the TLS session
capabilities, PT-TCP also allows for the re-use of the TLS based (PT-
TCP) session to allow either the NEA Client or the NEA Server to
trigger subsequent NEA exchanges.

4.1.4. NEA Data Transport Phase in PT-TCP

Once the PT-TCP session has been established, either the NEA Client
or the NEA Server can trigger a NEA data transaction (typically a
posture assessment). The initiator for the NEA data transaction
encapsulates the PB messages in a TLV as described in Section 5.1.

As PT-TCP is full-duplex (by the TLS design), it supports the full
capabilities of the PB-TNC state machine.

4.1.5. Entity Authentication using SASL in PT-TCP

Implementations may support entity authentication through the use of
SASL [RFC4422]. This section details the SASL profile for PT-TCP.

Typically, the PT-TCP initiator will also initiate the SASL exchange.
The responder presents a list of SASL mechanism it supports through
the use of the SASL-AUTH-MECH TLV. The initiator may request a list
of SASL authentication mechanisms by sending an empty list of
mechanisms in the SASL-AUTH-MECH TLV.

The initiator starts the authentication by sending a SASL-AUTH TLV
with the mech field containing the name of the mechanism it selects.
If the selected mechanism has an initial response then the client
includes that response in the auth-data field. If necessary the
responder sends a SASL-AUTH TLV with the auth-data field containing a
SASL challenge for the selected mechanism. The SASL-AUTH exchange
continues in this manner until the authentication completes upon
completion the responder sends a SASL-RESULT TLV. If the
authentication is successful the SASL-RESULT TLV contains an result
code of success. If the authentication fails the SASL-RESULT TLV
contains a result code indicating the reason for the failure. The
initiator may abort the exchange by sending a SASL-RESULT TLV with an

ABORT result code.

Implementations MUST provide the EXTERNAL SASL mechanism if the initiator is authenticated during the TLS establishment. Implementations MUST also support the PLAIN SASL mechanism.

4.1.5.1. Service Name

The service name for PT-TCP is "nea-pt-tcp".

4.1.5.2. Mechanism Negotiation

Mechanism Negotiation is performed using the SASL-AUTH-MECH TLV. The SASL-AUTH-MECH TLV contains the list of mechanisms supported by the responder. The initiator may send a SASL-AUTH-MECH TLV with an emptily list to request a list format from the responder.

4.1.5.3. Message Definition

The initiator starts authentication by sending a SASL-AUTH TLV indicating the sleeted mechanism. The initial message may contain an initial response if required by the selected mechanism. Subsequent challenges and response are carried within SASL-AUTH TLVs between the initiator and responder carrying the authentication data for the mechanism. The authentication outcome is communicated in a SASL-RESULT TLV containing a status code.

4.1.5.4. Authorization Identity

The nea-pt-tcp protocol does not make use of an authorization identity.

4.1.5.5. Aborting Authentication

The initiator may abort the authentication exchange by sending the SASL-AUTH-RESULT TLV with a status code of ABORT.

4.1.5.6. Security Layers

The NEA PT-TCP protocol always runs under the protection of TLS. SASL security layers are not used.

4.1.5.7. Multiple Authentications

Only one authentication may be in progress at any one time. Once an authentication completes, successfully on unsuccessfully, a new authentication may be initiated.

4.2. Tunnel EAP 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 when using a
   tunnel EAP method.

   When NEA data exchange is conducted in a tunnel EAP method, it
   typically consists of four phases:

   1. Establishment of EAP tunnel method: a secure and protected TLS
       channel is established between the Transport Client and Transport
       Server, after the Transport Server's identity has been
       authenticated and a shared secret encryption key is established
       between them.

   2. Entity authentication: during this phase, the NEA Client's
       Posture Transport Client's identity might be optionally
       authenticated, so appropriate posture assessment policy could be
       applied according to the authenticated entity. Typically, it is
       done via an inner EAP method or authentication exchanges within
       the protected tunnel. In addition, the identity could also be
       authenticated as part of the tunnel establishment instead (e.g.,
       the client sends a client certificate as part of the tunnel
       establishment).

   3. Posture assessment: the posture data are exchanged between the
       NEA Client's Posture Transport client and NEA Server's Posture
       Transport Server. The posture data is encapsulated in a TLV or
       TLV like type object, as described in Section 5.2.

   4. Conclusion phase: the result of the authentication and/or posture
       assessment is exchanged between the client and server, so they
       will have synchronized state. Optional cryptographic binding
       might be done to ensure both peers are involved in both the
       tunnel establishment and the inner method exchanges. Both sides
       are ready to tear down the tunnel and finish the EAP method.

   At the end of the tunnel EAP method, an EAP-Success or EAP-Failure
   will be sent by the EAP server to indicate the end of the EAP
   authentication, and the NAS will apply appropriate authorization
   policy based on the authentication and posture assessment result.

5. Packet Formats

   As there is no explicit authentication expected in the PB-PA message
   exchanges, no new inner EAP method is required; rather, the TLV

formats defined in existing EAP tunnel methods can be used to
encapsulate and transport PB-PA messages. Similarly, when using TLS
a TLV format can be defined to carry NEA data. This section
describes how NEA data can be carried in either a tunnel EAP method
or TLS.

5.1. PT-TCP transport Format

   This section defines a Type-Length-Value (TLV) encapsulation for
   carrying NEA data in a TLS channel. The TLS channel MUST be
   protected to carry NEA data using the encapsulation defined below.
   The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| R |            TLV Type        |             Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length             |            Data              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              Data                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

      R
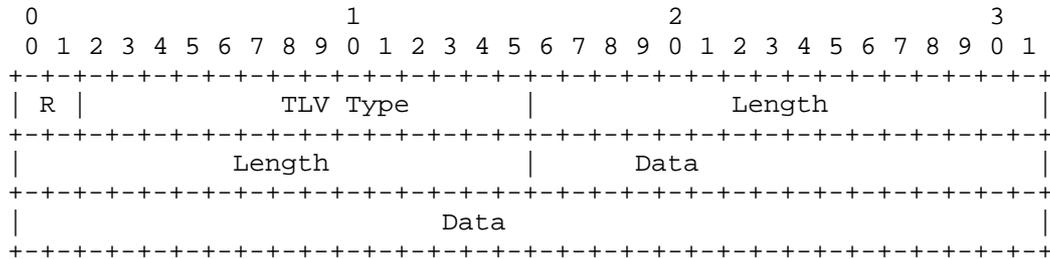
         Reserved, set to zero (0)

   TLV Type

      TLV Type Code. Allocated Types include:

      0 Reserved

      1 NEA TLV

      2 SASL-MECH TLV

      3 SASL-AUTH TLV

      4 SASL-RESULT TLV

   Length

      The length of the Data field in octets.

   Data

      Data according to the TLV type.

5.1.1. NEA TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| R |          TLV Type          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |         PB-TNC Header          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          PB-TNC Header         |        PB-PA Message...        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        PB-PA Message...                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


   R
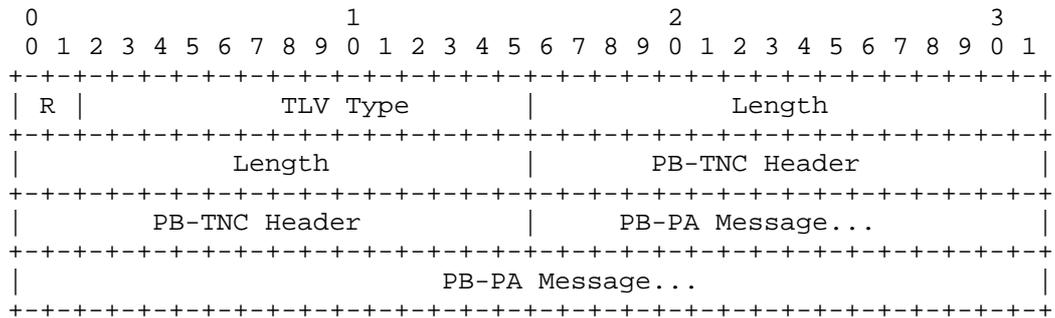
       Reserved, set to zero (0)

   TLV Type

      1 for NEA TLV

   Length

      The length of the Value field in octets.

   PB-TNC Header

      The PB-TNC encapsulation header as described in [RFC5793].

   PB-PA Message

      The message between the Posture Broker Client and Posture Broker
      Server as described in [RFC5793].

5.1.2. SASL-MECH TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| R |            TLV Type          |            Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |       Mech-Name-Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Mechanism Name                                   |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Mech-Name-Length           |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|              Mechanism Name                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
...
```
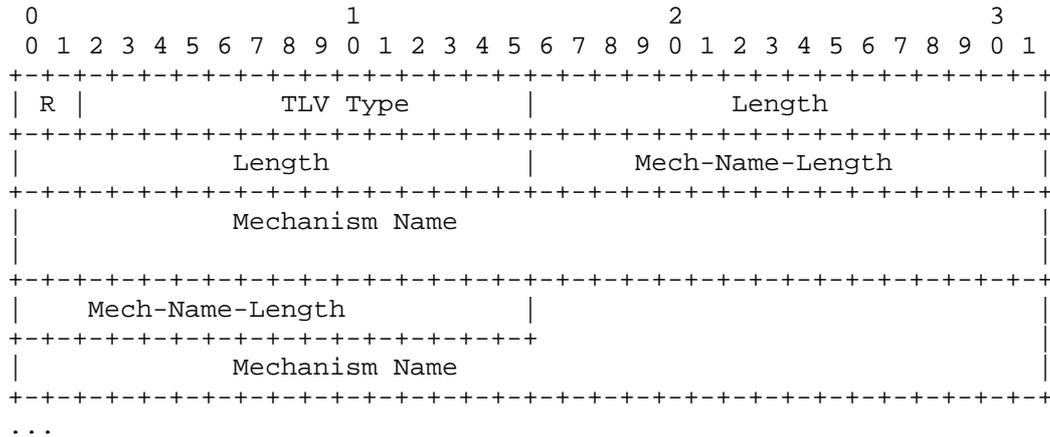
      The SASL-MECH TLV contains a list of supported SASL mechanisms. Each
      mechanism name consists of a name length followed by the name. The
      total length of the list is determined by the TLV length field.

      R

         Reserved, set to zero (0)

      TLV Type

         2 for SASL-MECH TLV

      Length

         The length of the Value field in octets. The value field contains
         the list of mechanism names.

   Mech-Name-Length

      Length of the mechanism name in bytes.

   Mech-Name

      SASL mechanism Name adhering to the rules defined in [RFC4422].

5.1.3. SASL-AUTH TLV


```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | R |           TLV Type        |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Length            |        Mech-Name-Length       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Mechanism Name                         |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                                                               |
   |                        Mechanism Data                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
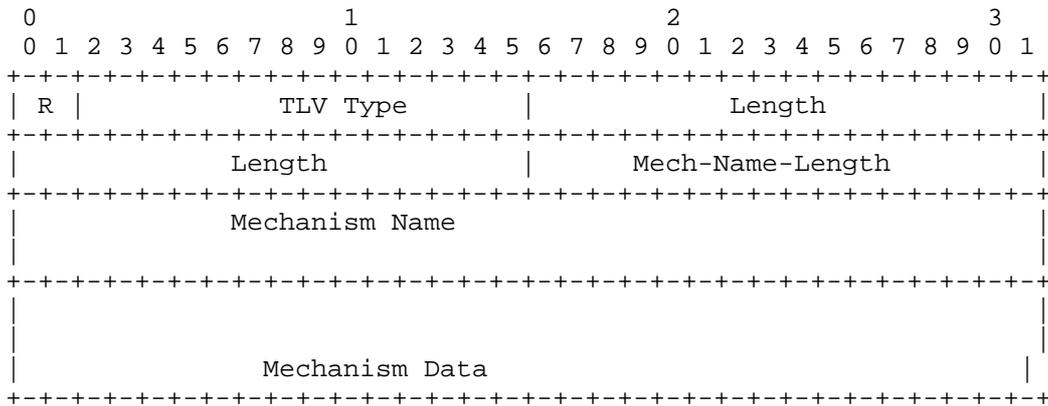

   The SASL-AUTH TLV contains data pertaining a SASL mechanism. The
   mechanism name is included in each SASL-AUTH TLV. The TLV is used by
   the initiator to select from a list of supported mechanisms provided
   by the responder. The initial response from the initiator may
   contain Mechanism Data containing the initial response. If the
   mechanism selected does not use an initial response then the
   mechanism data field is not included. The SASL-AUTH TLV is also used
   to communicate SASL mechanism data from the responder to the
   initiator.

   R

      Reserved, set to zero (0)

TLV Type

   3 for SASL-AUTH TLV

Length

   The length of the Value field in octets. The value field contains
   a mechanism name and optional mechanism data..


Mech-Name-Length

   Length of the mechanism name in bytes.

Mech-Name

   SASL mechanism Name adhering to the rules defined in [RFC4422].
   This is the mechanism selected for use by the initiator.

Mech-Name

   SASL mechanism data for named mechanism. This field may be
   omitted in the initial response from the initiator if the selected
   mechanism does not use an initial response.

5.1.4. SASL-RESULT TLV


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| R |          TLV Type         |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length            |          Result-Code          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
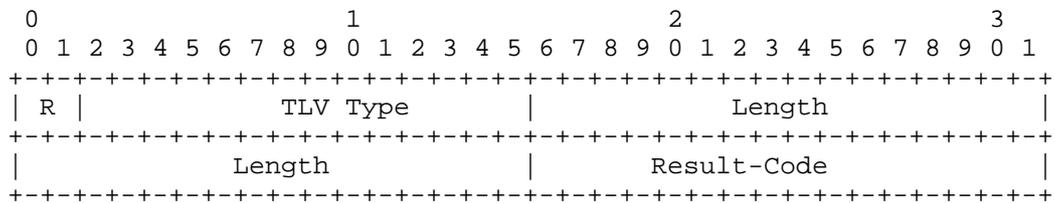

   The SASL-RESULT TLV contains the result of the SASL Exchange. A
   result code of 0 indicates success. Other result codes indicate some
   sort of failure. A result code of 1 indicates the exchange was
   aborted by the sender. A result code of 2 indicates a failure within
   the mechanism. Only the responder side of the conversation may send
   a successful result code. Either side may send a failure result code
   which terminates the current authentication conversation.

R

Reserved, set to zero (0)

TLV Type

4 for SASL-Result TLV

Length

The length of the Value field in octets. This field is set to 2.

Result Code

The value of the result code.

0 Success

1 Abort

2 Mechanism Failure

5.2. Using tunnel EAP to transport NEA data

This section describes the TLV encapsulation used in three
predominant tunnel EAP methods deployed today: PEAP, EAP-FAST and
TTLS. When using EAP tunnel methods, the tunnel MUST be protected.

5.2.1. Carrying NEA data in PEAP or EAP-FAST

As TLV format for PEAP and EAP-FAST are the same, the diagram below
shows how PB-PA messages can be encapsulated in the TLVs. Note
however that the type assignments when using PEAP versus EAP-FAST may
be different. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|R|          TLV Type          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PB-TNC Header                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PB-PA Message...                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
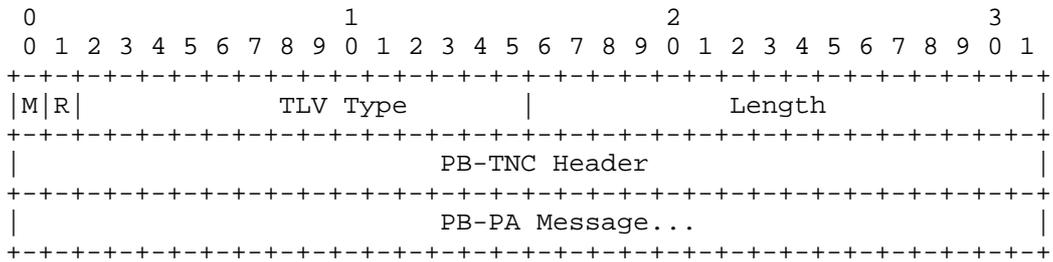
M

   0 Optional TLV

   1 Mandatory TLV

R

   Reserved, set to zero (0)

TLV Type

   The EAP-FAST NEA TLV type:

   TBD

Length

   The length of the Value field in octets.

PB-TNC Header

   The PB-TNC encapsulation header as described in [RFC5793].

PB-PA Message

   The message between the Posture Broker Client and Posture
   Broker Server as described in [RFC5793].

5.2.2. Carrying NEA data in TTLS

   The TTLS AVP Format to carry PB-PA messages is defined and described
   below. The fields are transmitted from left to right.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            AVP Code                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AVP Flags   |              AVP Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         PB-TNC Header                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                              |
|                       PB-PA-Message...                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
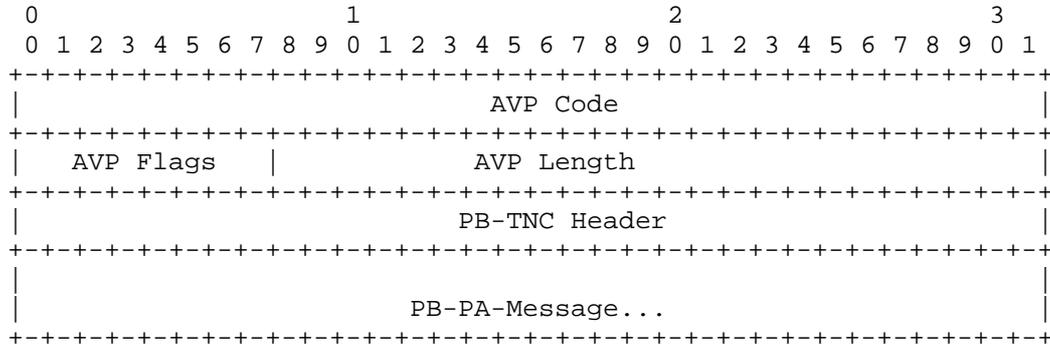
      AVP Code

         The TTLS NEA AVP type code:

         TBD

      AVP Flags

         The AVP flags are set to 0.

      AVP Length

         The length of the AVP in octets.

      PB-TNC Header

         The PB-TNC encapsulation header as described in [RFC5793].

      PB-PA Message

         The message between the Posture Broker Client and Posture
         Broker Server as described in [RFC5793].

6. Binding the PA exchange to the TLS Tunnel

   Some implementations of the NEA system allow for the external
   validation of the data collected and sent by the posture collector.
   In these cases, an external measurement agent (EMA) signs the data
   sent by the collector. In order to prevent posture data of the
   endpoint from being used on another machine, the TLS tunnel and the
   posture data signed by the EMA must be bound together. This is done
   using the "tls-unique" channel binding defined in RFC 5929 [RFC5929].
   The data from the first TLS Finished message sent on the most recent
   TLS connection handshake is included in the data signed by the EMA.
   The PA attributes used are specific to the EMA used by the posture
   collector.

   The "tls-unique" channel-binding data can be used whenever a TLS
   transport is provided, including TLS over TCP and TLS used in tunnel
   EAP methods. It is RECOMMENDED that posture collectors that support
   an EMA provide a PA attribute to carry the "tls-unique" channel
   binding data. The channel binding data MAY be combined with other
   data using a cryptographic hash or similar technique. The channel
   binding attribute MUST be signed by the EMA. Posture validators that
   receive channel binding data MUST verify that it is consistent with
   the channel binding data obtained from the server-side of the TLS
   connection.

7. Security Considerations

   The NEA TLV container carries network endpoint assessment information
   between the Posture Broker Client and the Posture Broker Server. As
   some of this data can be sensitive, TLS cipher suites that provide
   encryption are RECOMMENDED.

   To address the potential man-in-the-middle attack similar to the
   Asokan attack described in [I-D.salowey-nea-asokan] the channel
   binding mechanism defined in Section 6 SHOULD be used whenever an EMA
   is available to sign the posture data.

8. IANA Considerations

   IANA is requested to assign a TCP port number in the "Registered
   Port" range with the keyword "pt-tcp". This port will be the default
   port for PT-TCP defined in this document.

   IANA is requested to allocate a TLV type from the EAP-FAST TLV Type
   registry for carrying posture data as specified in Section 5.2.1.

IANA is requested to allocate a Diameter AVP code from the Diameter
AVP code registry for carrying posture data as specified in
Section 5.2.2.

This document defines a registry for TLV types to be carried within
PT-TCP, which may be assigned by Specification Required as defined in
[RFC2434]

9. Acknowledgements

The authors would like to recognize Susan Thomson, Syam Appala and
Subbu Srinivasan for providing input into this draft.

10. References

10.1. Normative References

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

   [RFC2434]   Narten, T. and H. Alvestrand, "Guidelines for Writing an
               IANA Considerations Section in RFCs", BCP 26, RFC 2434,
               October 1998.

   [RFC3748]   Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H.
               Levkowetz, "Extensible Authentication Protocol (EAP)",
               RFC 3748, June 2004.

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

   [RFC4493]   Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The
               AES-CMAC Algorithm", RFC 4493, June 2006.

   [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.

10.2. Informative References

   [I-D.salowey-nea-asokan]
               Salowey, J. and S. Hanna, "NEA Asokan Attack Analysis",
               draft-salowey-nea-asokan-00 (work in progress),

October 2010.

Appendix A. Evaluation Against NEA Requirements

   This section evaluates both the PT-TCP and EAP based protocols
   against the PT requirements defined in the NEA Overview and
   Requirements and PB-TNC specifications.

A.1. Evaluation Against Requirement C-1

   Requirement C-1 states:

   C-1 NEA protocols MUST support multiple round trips between the NEA
   Client and the NEA Server in a single assessment.

   PT-TCP meets this requirement. By using the TLS protocol over TCP,
   multiple roundtrips of TLS records and thus PT-TCP messages are
   allowed.

   Tunnel EAP meets this requirement. All available Tunnel EAP methods
   are based on the TLS design which allows for multiple round trips.

A.2. Evaluation Against Requirement C-2

   Requirement C-2 states:

   C-2 NEA protocols SHOULD provide a way for both the NEA Client and
   the NEA Server to initiate a posture assessment or reassessment as
   needed.

   PT-TCP meets this requirement. PT-TCP allows either the NEA Client
   or the NEA Server to initiate an assessment or reassessment.

   Tunnel EAP does not meet this requirement. The typical use case
   scenario for using a Tunnel EAP method is to service the layer 2
   network stack. In this use case, the endpoint would not have an IP
   address yet as it is requesting network access and thus would not be
   able to accept requests from the NEA Server. However, once network
   access has been granted, then yes, the NEA Client could receive
   (re)assessment requests from the NEA Server.

A.3. Evaluation Against Requirement C-3

   Requirement C-3 states:

   C-3 NEA protocols including security capabilities MUST be capable of
   protecting against active and passive attacks by intermediaries and

endpoints including prevention from replay based attacks.

PT-TCP meets this requirement. TLS includes mechanisms that provide strong cryptographic authentication, message integrity and confidentiality for NEA. In addition, to further mitigate man-in-the middle attacks, the use of channel binding at the PA layer must be used.

Tunnel EAP meets this requirement. All available Tunnel EAP methods are based on the TLS design which provide strong cryptographic authentication, message integrity and confidentiality for NEA. In addition, to further mitigate man-in-the middle attacks, the use of channel binding at the PA layer must be used.

A.4. Evaluation Against Requirement C-4

   Requirement C-4 states:

   C-4 The PA and PB protocols MUST be capable of operating over any PT protocol.

   This requirement is not applicable to PT, though the PT-TCP protocol is independent of both the PA and PB layer.

   This requirement is not applicable to PT, though the Tunnel EAP protocols are independent of both the PA and PB layer.

A.5. Evaluation Against Requirement C-5

   Requirement C-5 states:

   C-5 The selection process for NEA protocols MUST evaluate and prefer the reuse of existing open standards that meet the requirements before defining new ones. The goal of NEA is not to create additional alternative protocols where acceptable solutions already exist.

   As TLS is a widely used open standard, it should meet this requirement.

   As EMU is still in the early stages of standardizing a Tunnel EAP method, this specification reuses already widely deployed, published Tunnel EAP methods. Rather than defining a new Tunnel EAP method, this specification proposes to adopt already used ones and provides guidance for how new Tunnel EAP methods can meet this criteria to allow for NEA to use the method standardized by EMU at some future date.

A.6. Evaluation Against Requirement C-6

   Requirement C-6 states:

   C-6 NEA protocols MUST be highly scalable; the protocols MUST support
   many Posture Collectors on a large number of NEA Clients to be
   assessed by numerous Posture Validators residing on multiple NEA
   Servers.

   PT-TCP meets this requirement. As PT-TCP is a protocol to establish
   a protected channel by which NEA data can be transported, it is
   independent of the content of the data it is transporting and thus
   can allow for carrying batches of data to multiple Posture Validators
   or Posture Collectors.

   Tunnel EAP methods meet this requirement. As the Tunnel EAP methods
   define a protected transport channel that is independent of the
   content it transports, it can carry batches of data from and to
   multiple Posture Collectors and Posture Validators.

A.7. Evaluation Against Requirement C-7

   Requirement C-7 states:

   C-7 The protocols MUST support efficient transport of a large number
   of attribute messages between the NEA Client and the NEA Server.

   PT-TCP meets this requirement. The PT-TCP usurps 6 octets of
   overhead per PT-TCP message; a small overhead to the ability to carry
   very many PA-TNC attributes within a PB-TNC batch.

   The Tunnel EAP methods meet this requirements subject to the
   limitations of the underlying EAP protocol and encapsulation
   mechanisms. Note that a typical use case for the Tunnel EAP methods
   is that the assessments are brief and used for enabling network
   access; as such, it is not recommended to use Tunnel EAP methods to
   carry large amounts of attributes.

A.8. Evaluation Against Requirement C-8

   Requirement C-8 states:

   C-8 NEA protocols MUST operate efficiently over low bandwidth or high
   latency links.

   PT-TCP meets this requirement. As TLS was originally designed to
   work at the TCP layer, it has been proven to work well over either
   low bandwidth or high latency links.

EAP Tunnel methods meet this requirement. The underlying EAP
framework was designed and proven to work under constrained and low
latency links.

A.9. Evaluation Against Requirement C-9

Requirement C-9 states:

C-9 For any strings intended for display to a user, the protocols
MUST support adapting these strings to the user's language
preferences.

PT-TCP meets this requirement. The PT-TCP protocol does not define
messages intended for display to the user.

EAP Tunnel methods meet this requirement. The EAP Tunnel methods do
not define messages intended for display to the user.

A.10. Evaluation Against Requirement C-10

Requirement C-10 states:

C-10 NEA protocols MUST support encoding of strings in UTF-8 format.

PT-TCP meets this requirement. The PT-TCP protocol does not define
messages intended for display to the user.

EAP Tunnel methods meet this requirement. The EAP Tunnel methods do
not define messages intended for display to the user.

A.11. Evaluation Against Requirement C-11

Requirement C-11 states:

C-11 Due to the potentially different transport characteristics
provided by the underlying candidate PT protocols, the NEA Client and
the NEA Server MUST be capable of becoming aware of and adapting to
the limitations of the available PT protocol.

PT-TCP meets this requirement. The PT-TCP protocol uses TLS which is
designed to provide reliable transport that can adapt to constrained
or low bandwidth links.

EAP Tunnel methods meet this requirement. The EAP Tunnel methods are
based on TLS which is designed to provide reliable transport and have
been proven to adapt and work well under high latency or low
bandwidth conditions.

A.12. Evaluation Against Requirement PT-1

   Requirement PT-1 states:

   PT-1 The PT protocol MUST NOT interpret the contents of PB messages
   being transported, i.e., the data it is carrying must be opaque to
   it.

   PT-TCP meets this requirement. The PT-TCP protocol encapsulates PB
   messages in a TLV container without interpreting their contents.

   EAP Tunnel methods meet this requirement. The EAP Tunnel methods
   define encapsulations for carrying arbitrary data without
   interpreting their contents.

A.13. Evaluation Against Requirement PT-2

   Requirement PT-2 states:

   PT-2 The PT protocol MUST be capable of supporting mutual
   authentication, integrity, confidentiality, and replay protection of
   the PB messages between the Posture Transport Client and the Posture
   Transport Server.

   PT-TCP meets this requirement. The PT-TCP protocol uses TLS to
   provide mutual authentication, integrity, confidentiality, and replay
   protection.

   EAP Tunnel methods meet this requirement. The EAP Tunnel methods are
   based on TLS which is designed to provide mutual authentication,
   integrity, confidentiality, and replay protection.

A.14. Evaluation Against Requirement PT-3

   Requirement PT-3 states:

   PT-3 The PT protocol MUST provide reliable delivery for the PB
   protocol. This includes the ability to perform fragmentation and
   reassembly, detect duplicates, and reorder to provide in-sequence
   delivery, as required.

   PT-TCP meets this requirement. The PT-TCP protocol is designed to
   work over TCP which provides the fragmentation and reassembly
   services, detect duplicates and reorder messages if they arrive out
   of order.

   EAP Tunnel methods meet this requirement. The EAP Tunnel methods are
   based on the EAP framework which provides retransmissions, while

reordering and fragmentation are handled by the individual EAP Tunnel methods.

A.15. Evaluation Against Requirement PT-4

Requirement PT-4 states:

PT-4 The PT protocol SHOULD be able to run over existing network access protocols such as 802.1X and IKEv2.

PT-TCP does NOT meet this requirement as it is designed to operate over TCP.

EAP Tunnel methods meet this requirement. The EAP Tunnel methods are based on EAP which has been enabled on both 802.1X and IKEv2.

A.16. Evaluation Against Requirement PT-5

Requirement PT-5 states:

PT-5 The PT protocol SHOULD be able to run between a NEA Client and NEA Server over TCP or UDP (similar to Lightweight Directory Access Protocol (LDAP))

PT-TCP meets this requirement. The PT-TCP protocol is designed to operate over a TCP connection.

EAP Tunnel methods do NOT meet this requirement. The EAP Tunnel methods are designed to work pre-network admission and thus are not able to communicate at the IP layer.

A.17. Evaluation Against Requirement PT-6

Requirement PT-6 states:

PT-6 The PT protocol MUST be connection oriented; it MUST support confirmed initiation and close down.

PT-TCP meets this requirement. The PT-TCP protocol is designed to operate over a TCP connection which is connection oriented and supports initiation and tear down of the connection.

EAP Tunnel methods meet this requirement. The EAP Tunnel methods are based on EAP which provides both initiation and shutdown.

A.18. Evaluation Against Requirement PT-7

   Requirement PT-7 states:

   PT-7 The PT protocol MUST be able to carry binary data.

   PT-TCP meets this requirement. The PT-TCP protocol is capable of
   carrying binary data.

   EAP Tunnel methods meet this requirement. The EAP Tunnel methods are
   capable of carrying binary data.

A.19. Evaluation Against Requirement PT-8

   Requirement PT-8 states:

   PT-8 The PT protocol MUST provide mechanisms for flow control and
   congestion control.

   PT-TCP meets this requirement. The PT-TCP protocol operates over TCP
   which provides flow control and congestion control.

   EAP Tunnel methods meet this requirement. The EAP Tunnel methods are
   based on EAP which, by use of the half-duplex, round-robin message
   exchange, flow and congestion control are provided.

A.20. Evaluation Against Requirement PT-9

   Requirement PT-9 states:

   PT-9 The PT protocol specifications MUST describe the capabilities
   that they provide for and limitations that they impose on the PB
   protocol (e.g. half/full duplex, maximum message size).

   PT-TCP meets this requirement. This specification has provided the
   required information.

   EAP Tunnel methods meet this requirement. This specification has
   provided the required information.

Authors' Addresses

    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

    Hao Zhou
    Cisco Systems
    4125 Highlander Parkway
    Richfield, OH 44286
    US

    Email: hzhou@cisco.com