PT-EAP: Posture Transport (PT) Protocol For EAP Tunnel Methods
draft-ietf-nea-pt-eap-02

Abstract

   This document specifies PT-EAP, an EAP based Posture Transport (PT)
   protocol designed to be used only inside a TLS protected tunnel
   method. The document also describes the intended applicability of
   PT-EAP.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on November 16, 2012.

Copyright Notice

Table of Contents

1. Introduction

   This document specifies PT-EAP, a Posture Transport (PT) protocol
   protected by an EAP TLS-based tunnel outer method. The PT protocol
   in the NEA architecture is responsible for transporting PB-TNC
   batches (often containing PA-TNC [RFC5792] attributes) across the
   network between the NEA Client and NEA Server. The PT-EAP protocol
   must be protected by an outer EAP TLS-based tunnel to ensure the
   exchanged messages are protected from a variety of threats from
   hostile intermediaries.

   NEA protocols are intended to be used both 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 types of PT protocols are needed. One type of PT
   operates after the endpoint has an assigned IP address, layering on
   top of the IP protocol to carry a NEA exchange. The other type of
   PT, PT-TLS [I-D.ietf-nea-pt-tls], operates before the endpoint gains
   any access to the IP network. This specification defines PT-EAP, the
   PT protocol used to assess endpoints before they gain access to the
   network.

   PT-EAP is an inner EAP [RFC3748] method designed to be used under a
   protected tunnel such as TEAP [I-D.ietf-emu-eap-tunnel-method], EAP-
   FAST [RFC4851] or EAP-TTLS [RFC5281].

1.1. Prerequisites

   This document does not define an architecture or reference model.
   Instead, it defines a protocol that works within the reference model
   described in the NEA Requirements specification [RFC5209]. The
   reader is assumed to be thoroughly familiar with that document.

1.2. Message Diagram Conventions

   This specification defines the syntax of PT-EAP 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.

   Descriptions of 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. Terminology

This document reuses many terms defined in the NEA Requirements
document [RFC5209], such as Posture Transport Client and Posture
Transport Server. The reader is assumed to have read that document
and understood it.

When defining the PT-EAP method, this specification does not use the
terms "EAP peer" and "EAP authenticator". Instead, it uses the terms
"NEA Client" and "NEA Server" since those are considered to be more
familiar to NEA WG participants. However, these terms are equivalent
for the purposes of these specifications. The part of the NEA Client
that terminates PT-EAP (generally in the Posture Transport Client) is
the EAP peer for PT-EAP. The part of the NEA Server that terminates
PT-EAP (generally in the Posture Transport Server) is the EAP
authenticator for PT-EAP.

1.4. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

1.5. 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 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. Use of PT-EAP

PT-EAP is designed to encapsulate PB-TNC batches in a simple EAP
method that can be carried within EAP tunnel methods. The EAP tunnel
methods provide confidentiality and message integrity, so PT-EAP does
not have to do so. Therefore, PT-EAP MUST only be used inside an EAP
TLS-based tunnel method that provides strong cryptographic
authentication (possibly server only), message integrity and
confidentiality services.

3. Definition of PT-EAP

The PT-EAP protocol operates between a Posture Transport Client and a
Posture Transport Server, allowing them to send PB-TNC batches to
each other over an EAP tunnel method. When PT-EAP is used, the

Posture Transport Client in the NEA reference model acts as an EAP
peer (terminating the PT-EAP method on the endpoint) and the Posture
Transport Server acts as an EAP authenticator (terminating the PT-EAP
method on the NEA Server).

This section describes and defines the PT-EAP method. First, it
provides a protocol overview and a flow diagram. Second, it
describes specific features like version negotiation. Third, it
gives a detailed packet description. Finally, it describes how the
tls-unique channel binding [RFC5929] may be used to PA-TNC exchanges
to the EAP tunnel method, defeating MITM attacks such as the Asokan
attack [Asokan].

3.1. Protocol Overview

PT-EAP has two phases that follow each other in strict sequence:
negotiation and data transport.

The PT-EAP method begins with the negotiation phase. The NEA Server
starts this phase by sending an PT-EAP Start message: an EAP Request
message of type PT-EAP with the S (Start) flag set. The NEA Server
also sets the Version field as described in Section 3.2. This is the
only message in the negotiation phase.

The data transport phase is the only phase of PT-EAP where PB-TNC
batches are allowed to be exchanged. This phase always starts with
the NEA Client sending a PB-TNC batch to the NEA Server. The NEA
Client and NEA Server then engage in a round-robin exchange with one
PB-TNC batch in flight at a time. The data transport phase always
ends with an EAP Response message from the NEA Client to the NEA
Server. This message may be empty (not contain any data) if the NEA
Server has just sent the last PB-TNC batch in the PB-TNC exchange.

At the end of the PT-EAP method, the NEA Server will indicate success
or failure to the EAP tunnel method. Some EAP tunnel methods may
provide explicit confirmation of inner method success; others may
not. This is out of scope for the PT-EAP method specification.
Successful completion of PT-EAP does not imply successful completion
of the overall authentication nor does PT-EAP failure imply overall
failure. This depends on the administrative policy in place.

The NEA Server and NEA Client may engage in an abnormal termination
of the PT-EAP exchange at any time by simply stopping the exchange.
This may also require terminating the EAP tunnel method, depending on
the capabilities of the EAP tunnel method.

The NEA Server and NEA Client MUST follow the protocol sequence
described in this section.

3.2. Version Negotiation

PT-EAP version negotiation takes place in the first PT-EAP message
sent by the NEA Server (the Start message) and the first PT-EAP
message sent by the NEA Client (the response to the Start message).
The NEA Server MUST set the Version field in the Start message to the
maximum PT-EAP version that the NEA Server supports and is willing to
accept.

The NEA Client chooses the PT-EAP version to be used for the exchange
and places this value in the Version field in its response to the
Start message. The NEA Client SHOULD choose the value sent by the
NEA Server if the NEA Client supports it. However, the NEA Client
MAY set the Version field to a value less than the value sent by the
NEA Server (for example, if the NEA Client only supports lesser PT-
EAP versions). If the NEA Client only supports PT-EAP versions
greater than the value sent by the NEA Server, the NEA Client MUST
abnormally terminate the EAP negotiation.

If the version sent by the NEA Client is not acceptable to the NEA
Server, the NEA Server MUST terminate the PT-EAP session immediately.
Otherwise, the version sent by the NEA Client is the version of PT-
EAP that MUST be used. Both the NEA Client and the NEA Server MUST
set the Version field to the chosen version number in all subsequent
PT-EAP messages in this exchange.

This specification defines version 1 of PT-EAP. Version 0 is
reserved and MUST never be sent. New versions of PT-EAP (values 2-7)
may be defined by Standards Action, as defined in [RFC5226].

3.3. PT-EAP Message Format

This section provides a detailed description of the fields in an PT-
EAP message. For a description of the diagram conventions used here,
see Section 1.2. Since PT-EAP is an EAP method, the first four
fields in each message are mandated by and defined in EAP.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Code      |   Identifier  |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      | Flags | Ver |          Data ...               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
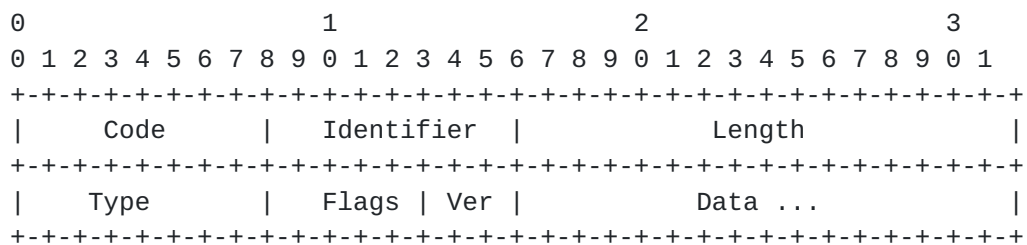
Code

   The Code field is one octet and identifies the type of the EAP
   message. The only values used for PT-EAP are:

1 Request

2 Response

Identifier

   The Identifier field is one octet and aids in matching Responses
   with Requests.

Length

   The Length field is two octets and indicates the length in octets
   of this PT-EAP message, starting from the Code field.

Type

   TBD

Flags

                        +-+-+-+-+-+
                        |S R R R R|
                        +-+-+-+-+-+

S: Start

   Indicates the beginning of an PT-EAP exchange. This flag MUST be
   set only for the first message from the NEA Server. If the S flag
   is set, the EAP message MUST NOT contain Data.

R: Reserved

   This flag MUST be set to 0 and ignored upon receipt.

Version

   This field is used for version negotiation, as described in
   Section 3.2.

Data

   Variable length data. The length of the Data field in a
   particular PT-EAP message may be determined by subtracting the
   length of the PT-EAP header fields from the value of the two octet
   Length field.

3.4. Preventing MITM Attacks with Channel Bindings

   As described in the NEA Asokan Attack Analysis [16], 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. 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, which measures
   and attests to the configuration of the endpoint.

   To protect against NEA Asokan attacks, it is necessary for the
   Posture Broker on an EMA-equipped endpoint to pass the tls-unique
   channel binding [RFC5929] for PT-EAP's tunnel method to the EMA.
   This value can then be included in the EMA's attestation so that the
   Posture Validator responsible may then confirm that the value matches
   the tls-unique channel binding for its end of the tunnel. If the
   tls-unique values of the NEA Client and NEA Server match and this is
   confirmed by the EMA, then the posture sent by the EMA (and thus the
   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, an attack has
   been detected and the Posture Validator SHOULD fail its verification.

4. Security Considerations

   This section discusses the major threats and countermeasures provided
   by PT-EAP. As discussed throughout the document, the PT-EAP method
   is designed to run inside an EAP tunnel method which is capable of
   protecting the PT-EAP protocol from many threats. Since the EAP
   tunnel method will be specified separately, this section describes
   the considerations on the EAP tunnel method but do not evaluate its
   ability to meet those requirements. The security considerations and

requirements for NEA can be found in [RFC5209]. .

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-EAP protocol. The following sub-sections discuss the trust properties associated with each portion of the NEA reference model directly involved with the processing of the PT-EAP protocol flowing inside an EAP tunnel.

4.1.1. Posture Transport Client

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

o Not to observe, fabricate or alter the contents of the PB-TNC batches received from the network

o Not to observe, fabricate or alter the PB-TNC batches passed down from the Posture Broker Client for transmission on the network

o Transmit on the network any PB-TNC batches passed down from the Posture Broker Client

o Deliver properly security protected messages received from the network that are destined for the Posture Broker Client

o Provide configured security protections (e.g. authentication, integrity and confidentiality) for the Posture Broker Client's PB-TNC batches sent on the network

o Expose the authenticated identity of the Posture Transport Server to the Posture Transport Client.

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

o 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:

o Not send malicious traffic intending to harm (e.g. denial of
   service) the Posture Transport Server

o Not to intentionally send malformed messages to cause processing
   problems for the Posture Transport Server

o Not to send invalid or incorrect responses to messages (e.g.
   errors when no error is warranted)

o Not to ignore or drop messages causing issues for the protocol
   processing

o Verify the security protections placed upon messages received from
   the network to ensure 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:

o Not to observe, fabricate or alter the contents of the PB-TNC
   batches received from the network

o Not to observe, fabricate or alter the PB-TNC batches passed down
   from the Posture Broker Server for transmission on the network

o Transmit on the network any PB-TNC batches passed down from the
   Posture Broker Server

o Deliver properly security protected messages received from the
   network that are destined for the Posture Broker Server

o Provide configured security protections (e.g. authentication,
   integrity and confidentiality) for the Posture Broker Server's
   messages sent on the network

o Expose the authenticated identity of the Posture Transport Client
   to the Posture Transport Server

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

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

o Not send malicious traffic intending to harm (e.g. denial of
   service) the Posture Transport Server

o Not to send malformed messages

o Not to send invalid or incorrect responses to messages (e.g.
   errors when no error is warranted)

o Not to ignore or drop messages causing issues for the protocol
   processing

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

4.2. Security Threats and Countermeasures

Beyond the trusted relationships assumed in Section 4.1, the PT-EAP
EAP method faces a number of potential security attacks that could
require security countermeasures.

Generally, the PT protocol is responsible for providing strong
security protections for all of the NEA protocols so any threats to
PT's ability to protect NEA protocol messages could be very damaging
to deployments. For the PT-EAP method, most of the cryptographic
security is provided by the outer EAP tunnel method and PT-EAP is
encapsulated within the protected tunnel. Therefore, this section
highlights the cryptographic requirements that need to be met by the
EAP tunnel method carrying PT-EAP in order to meet the NEA PT
requirements.

Once the message is delivered to the Posture Broker Client or Posture
Broker Server, the posture brokers are trusted to properly safely
process the messages.

4.2.1. Message Theft

When PT-EAP 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 easing the ability of 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-EAP is housed in an EAP tunnel
method that does not provide confidentiality protection, an adversary
could observe the PA-TNC attributes included in the PB-TNC batch and
determine that the endpoint is lacking patches, or particular sub-
networks have more lenient policies.

In order to protect against NEA assessment message theft, the EAP
tunnel method carrying PT-EAP must provide strong cryptographic
authentication, integrity and confidentiality protection. The use of
bi-directional authentication in the EAP tunnel method carrying PT-
EAP ensures that only properly authenticated and authorized parties
may be involved in an assessment message exchange. When PT-EAP is
carried within a cryptographically protected EAP tunnel method like
EAP-FAST or EAP-TTLS, all of the PB-TNC and PA-TNC protocol messages
contents are hidden from potential theft by intermediaries lurking on
the network.

4.2.2. Message Fabrication

Attackers on the network or present within the NEA system could
introduce fabricated PT-EAP 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-
EAP error codes in order to disrupt communications.

The EAP tunnel method carrying an PT-EAP method needs to provide
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.

4.2.3. Message Modification

This attack could allow an active attacker capable of intercepting a
message to modify a PT-EAP message or transported PA-TNC attribute to
a desired value to ease the compromise of 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-EAP method leverages the EAP tunnel method (e.g. EAP-TTLS) to
provide strong authentication and integrity protections as a
countermeasure to this threat. The bi-directional 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 protections (hashing) offered by EAP-TTLS allows the
PT-EAP message recipients to detect message alterations by other
types of network based adversaries. Because PT-EAP does not itself
provide explicit integrity protection for the PT-EAP payload, an EAP

tunnel method that offers strong integrity protection is required to
mitigate this threat.

4.2.4. Denial of Service

A variety of types of denial of service attacks are possible against
the PT-EAP if the message exchange 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-EAP method primarily relies on the outer EAP tunnel method to
provide strong authentication (at least of one party) and deployers
are expected to leverage other EAP methods to authenticate the other
party (typically the client) within the protected tunnel. The use of
a protected bi-directional authentication will prevent unauthorized
parties from participating in a PT-EAP exchange.

After the cryptographic authentication by the EAP tunnel method, the
session can be encrypted and hashed to prevent 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.

4.2.5. NEA Asokan Attacks

As described in Section 3.4. and in the NEA Asokan Attack Analysis
[Asokan], 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 3.4 and the NEA Asokan Attack Analysis
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-EAP implementers may not know whether an
EMA will be used with their implementation. Therefore, PT-EAP
implementers SHOULD support these countermeasures 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.

4.3. Candidate EAP Tunnel Method Protections

   This section discusses how PT-EAP is used within various EAP tunnel
   methods to meet the PT requirements from section Section 5.

   EAP-FAST [RFC4851] and EAP-TTLS [RFC5281] make use of TLS [RFC5246]
   to protect the transport of information between the NEA Client and
   NEA Server. Each of these EAP tunnel methods has two phases. In the
   first phase, a TLS tunnel is established between NEA Client and NEA
   Server. In the second phase, the tunnel is used to pass other
   information. PT-EAP requires that establishing this tunnel include
   at least an authentication of the NEA Server by the NEA Client.

   The phase two dialog may include authentication of the user by doing
   other EAP methods or in the case of TTLS by using non-EAP
   authentication dialogs. PT-EAP is also carried by the phase two
   tunnel allowing the NEA assessment to be within an encrypted and
   integrity protected transport.

   With all these methods (e.g. TEAP [I-D.ietf-emu-eap-tunnel-method],
   EAP-FAST [RFC4851] and EAP-TTLS [RFC5281]), a cryptographic key is
   derived from the authentication that may be used to secure later
   transmissions. Each of these methods employs at least a NEA Server
   authentication using an X.509 certificate. Within each EAP tunnel
   method will exist a set of inner EAP method (or an equivalent using
   TLVs if inner methods aren't directly supported.) These inner
   methods may perform additional security handshakes including more
   granular authentications or exchanges of integrity information (such
   as PT-EAP.) At some point after the conclusion of each inner EAP
   method, some of the methods will export the established secret keys
   to the outer tunnel method. It's expected that the outer method will
   cryptographically mix these keys into any keys it is currently using
   to protect the session and perform a final operation to determine
   whether both parties have arrived at the same mixed key. This
   cryptographic binding of the inner method results to the outer
   methods keys is essential for detection of conventional (non-NEA)
   Asokan attacks.

4.4. Security Claims for PT-EAP as per RFC3748

   This section summarizes the security claims for this specification,
   as required by RFC3748 Section 7.2:

            Auth. mechanism: None
            Ciphersuite negotiation: No
            Mutual authentication: No
            Integrity protection: No
            Replay protection: No
            Confidentiality: No
            Key derivation: No
            Key strength: N/A
            Dictionary attack resistant: N/A
            Fast reconnect: No
            Crypt. binding: N/A
            Session independence: N/A
            Fragmentation: No
            Channel binding: No

5. Requirements for EAP Tunnel Methods

   Because the PT-EAP inner method described in this specification
   relies on the outer EAP tunnel method for a majority of its security
   protections, this section reiterates the PT requirements that MUST be
   met by the IETF standard EAP tunnel method for use with PT-EAP.

   There is no mandatory to implement EAP tunnel based method defined in
   this draft since there exists no such standard method. At the time
   of this writing, the IETF EAP Method Unification (EMU) working group
   is working on standardizing on TEAP [I-D.ietf-emu-eap-tunnel-method]
   as the EAP tunnel method that will satisfy NEA's requirements.

   The security requirements described in this specification MUST be
   implemented in any product claiming to be PT-EAP compliant. The
   decision of whether a particular deployment chooses to use these
   protections is a deployment issue. A customer may choose to avoid
   potential deployment issues or performance penalties associated with
   the use of cryptography when the required protection has been
   achieved through other mechanisms (e.g. physical isolation). If
   security mechanisms may be deactivated by policy, an implementation
   SHOULD offer an interface to query how a message will be (or was)
   protected by PT so higher layer NEA protocols can factor this into
   their decisions.

   RFC 5209 [RFC5209] includes the following requirement that is to be
   applied during the selection of the EAP tunnel method(s) used in
   conjunction with PT-EAP:

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.

Note that mutual authentication could be achieved by a combination of
a strong authentication of one party (e.g. server authentication
while establishing the TLS-based tunnel) by the EAP tunnel method in
conjunction with a second authentication of the other party (e.g.
client authentication inside the protected tunnel) by another EAP
method running prior to PT-EAP.

Having the Posture Transport Client always authenticate the Posture
Transport Server provides assurance to the NEA Client that the NEA
Server is authentic (not a rogue or MiTM) prior to disclosing secret
or potentially privacy sensitive information about what is running or
configured on the endpoint. However the NEA Server's policy may
allow for the delay of the authentication of the NEA Client until a
suitable protected channel has been established allowing for non-
cryptographic NEA Client credentials (e.g. username/password) to be
used. Whether the communication channel is established with mutual
or server-side only authentication, the resulting channel needs to
provide strong integrity and confidentiality protection to its
contents. These protections are to be bound to at least the
authentication of the NEA Server by the NEA Client, so the session is
cryptographically bound to a particular authentication event.

The EAP tunnel method carrying PT-EAP MUST provide strong
cryptographic authentication, integrity and confidentiality
protection to protect against NEA assessment message theft as
described in Section 4.2.1. The cryptographically protected EAP
tunnel ensures that all of the PA-TNC and PB-TNC protocol messages
are hidden from intermediaries wanting to steal NEA data.

To support countermeasures against NEA Asokan attacks as described in
Section 3.4, the EAP Tunnel Method used with PT-EAP will need to
support the tls-unique channel binding. This should not be a high
bar since all EAP tunnel methods currently support this but not all
implementations of those methods may do so.

6. **Privacy Considerations**

The role of PT-EAP is to act as a secure transport for PB-TNC over a
network before the endpoint has been admitted to the network. As a
transport protocol, PT-EAP does not directly utilize or require
direct knowledge of any personally identifiable information (PII).
 PT-EAP will typically be used in conjunction with other EAP methods

that provide for the user authentication (if bi-directional
authentication is used), so the user's credentials are not directly
seen by the PT-EAP inner method. Therefore, the Posture Transport
Client and Posture Transport Server's implementation of PT-EAP MUST
NOT observe the contents of the carried PB-TNC batches that could
contain PII carried by PA-TNC or PB-TNC.

While PT-EAP does not provide cryptographic protection for the PB-TNC
batches, it is designed to operate within an EAP tunnel method that
provides strong authentication, integrity and confidentiality
services. Therefore, it is important for deployers to leverage these
protections in order to prevent disclosure of PII potentially
contained within PA-TNC or PB-TNC within the PT-EAP payload.

7. IANA Considerations

This section provides guidance to the Internet Assigned numbers
Authority (IANA) regarding registration of values related to the PT-
EAP protocol, in accordance with BCP 26 [RFC2434]

The EAP Method type for PT-EAP needs to be assigned.

This document also defines one new IANA registry: PT-EAP Versions.
This section explains how this registry works. Because only eight
(8) values are available in this registry, a high bar is set for new
assignments. The only way to register new values in this registry is
through Standards Action (via an approved Standards Track RFC).

7.1. Registry for PT-EAP Versions

The name for this registry is "PT-EAP Versions". Each entry in this
registry includes a decimal integer value between 1 and 7 identifying
the version, and a reference to the RFC where the version is defined.

The following entries for this registry are defined in this document.
Once this document becomes an RFC, they will become the initial
entries in the registry for PT-EAP Versions. Additional entries to
this registry are added by Standards Action, as defined in RFC 5226
[RFC5226].

```
+-------+----------------------------+
| Value |   Defining Specification   |
+-------+----------------------------+
|   1   | RFC # Assigned to this I-D |
+-------+----------------------------+
```

8. Acknowledgements

Thanks to the Trusted Computing Group for contributing the initial text upon which this document was based.

The authors of this draft would like to acknowledge the following people who have contributed to or provided substantial input on the preparation of this document or predecessors to it: Amit Agarwal, Morteza Ansari, Diana Arroyo, Stuart Bailey, Boris Balacheff, Uri Blumenthal, Gene Chang, Scott Cochrane, Pasi Eronen, Aman Garg, Sandilya Garimella, David Grawrock, Stephen Hanna, Thomas Hardjono, Chris Hessing, Ryan Hurst, Hidenobu Ito, John Jerrim, Meenakshi Kaushik, Greg Kazmierczak, Scott Kelly, Bryan Kingsford, PJ Kirner, Sung Lee, Lisa Lorenzin, Mahalingam Mani, Bipin Mistry, Seiji Munetoh, Rod Murchison, Barbara Nelson, Kazuaki Nimura, Ron Pon, Ivan Pulleyn, Alex Romanyuk, Ravi Sahita, Chris Salter, Mauricio Sanchez, Joseph Salowey, Dean Sheffield, Curtis Simonson, Jeff Six, Ned Smith, Michelle Sommerstad, Joseph Tardo, Lee Terrell, Susan Thomson, Chris Trytten, and John Vollbrecht.

This document was prepared using template-bare-05.xml.

9. References

9.1. Normative References

[IEEE]   IEEE Std. 802.1X-2004, "LAN/MAN Standards Committee of the IEEE Computer Society, Standard for Local and Metrolpolitan Area Network - Port Based Network Access Control", December 2004, <http://standards.ieee.org/getieee802/download/802.1X-2010.pdf>.

[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.

[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.

   [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.

9.2. Informative References

   [Asokan]   Asokan, N., Niemi, V., Nyberg, K., and Nokia Research
              Center, Finland, ""Man in the Middle Attacks in Tunneled
              Authentication Protocols"", Nov 2002,
              <http://eprint.iacr.org/2002/163.pdf>.

   [I-D.ietf-emu-eap-tunnel-method]
              Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna,
              "Tunnel EAP Method (TEAP) Version 1",
              draft-ietf-emu-eap-tunnel-method-02 (work in progress),
              March 2012.

   [I-D.ietf-emu-eaptunnel-req]
              Zhou, H., Salowey, J., Hoeper, K., and S. Hanna,
              "Requirements for a Tunnel Based EAP Method",
              draft-ietf-emu-eaptunnel-req-09 (work in progress),
              December 2010.

   [I-D.ietf-nea-pt-tls]
              Sangster, P., Cam-Winget, N., and J. Salowey, "PT-TLS: A
              TCP-based Posture Transport (PT) Protocol",
              draft-ietf-nea-pt-tls-04 (work in progress), May 2012.

   [I-D.salowey-nea-asokan]
              Salowey, J. and S. Hanna, "NEA Asokan Attack Analysis",
              draft-salowey-nea-asokan-01 (work in progress),
              March 2012.

   [RFC3478] Leelanivas, M., Rekhter, Y., and R. Aggarwal, "Graceful
              Restart Mechanism for Label Distribution Protocol",
              RFC 3478, February 2003.

   [RFC4851] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "The
              Flexible Authentication via Secure Tunneling Extensible
              Authentication Protocol Method (EAP-FAST)", RFC 4851,
              May 2007.

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

   [RFC5216]   Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS
               Authentication Protocol", RFC 5216, March 2008.

   [RFC5281]   Funk, P. and S. Blake-Wilson, "Extensible Authentication
               Protocol Tunneled Transport Layer Security Authenticated
               Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.

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

   [RFC5996]   Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen,
               "Internet Key Exchange Protocol Version 2 (IKEv2)",
               RFC 5996, September 2010.

   [TNC-Binding]
               Trusted Computing Group, ""TNC IF-T: Binding to TLS"",
               May 2009, <http://www.trustedcomputinggroup.org/files/
               resource_files/51F0757E-1D09-3519-AD63B6FD099658A6/
               TNC_IFT_TLS_v1_0_r16.pdf>.

Authors' Addresses

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

   Email: ncamwing@cisco.com

   Paul Sangster
   Symantec Corporation
   6825 Citrine Drive
   Carlsbad, CA 92009
   USA

   Email: paul_sangster@symantec.com