Network Working Group
Internet Draft
Intended status: Proposed Standard
Expires: May 2010

S. Hanna
Juniper Networks
P. Sangster
Symantec Corporation

March 3, 2011

PT-EAP: Posture Transport (PT) Protocol For EAP Tunnel Methods
draft-hanna-nea-pt-eap-01.txt

Abstract

This document specifies PT-EAP, a Posture Broker Protocol compatible with the Trusted Computing Group's IF-T Protocol Bindings for Tunneled EAP Methods (also known as EAP-TNC). The document then evaluates PT-EAP against the requirements defined in the NEA Requirements and PB-TNC specifications.

Table of Contents

1. Introduction

This document specifies PT-EAP, a Posture Transport Protocol (PT) compatible with the Trusted Computing Group's IF-T Protocol Bindings for Tunneled EAP Methods (also known as EAP-TNC) [10]. The document then evaluates PT-EAP against the requirements defined in the NEA Requirements [7] and PB-TNC specifications [4].

The PT protocol in the NEA architecture is responsible for transporting PB-TNC batches (often containing PA-TNC [3] attributes) across the network between the NEA Client and NEA Server. The PT protocol also offers strong security protections 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 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 comprised of two related protocols, an outer EAP tunnel method (not defined in this specification) and an inner EAP method that carries the NEA assessment inside the protections of the outer EAP tunnel method. This specification uses the term PT-EAP to refer to both collectively. The inner EAP method is based upon a method called EAP-TNC, which is part of the Trusted Computing Group's TNC architecture and standards. This specification defines the EAP-TNC inner EAP method, while allowing the outer EAP tunnel method to be specified in another specification (possibly defined by another IETF WG). The reason to define PT-EAP as including both the outer EAP tunnel method and the inner EAP method is because both are required to meet the PT requirements.

EAP-TNC is designed to operate as an inner EAP [8] method over an EAP tunnel method that meets the Requirements for a Tunnel Based EAP Method [15]. PT-EAP therefore can operate over a number of existing access protocols that support EAP for authentication. Some examples of such access protocols include 802.1X [5] for wired and wireless networks and IKEv2 [13] for establishing VPNs over IP networks.

This document defines a standard EAP inner method called EAP-TNC. It also shows how EAP-TNC may be carried over two existing EAP tunnel EAP methods: EAP-FAST [12] and EAP-TTLS [14].

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 [7]. The reader is assumed to be thoroughly familiar with that document. No familiarity with Trusted Computing Group (TCG) specifications is assumed.

1.2. Message Diagram Conventions

This specification defines the syntax of EAP-TNC 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 [7], such as Posture Transport Client and Posture Transport Server. The reader is assumed to have read that document and understood it.

When defining the EAP-TNC 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 EAP-TNC (generally in the Posture Transport Client) is the EAP peer for EAP-TNC. The part of the NEA Server that terminates EAP-TNC (generally in the Posture Transport Server) is the EAP authenticator for EAP-TNC.

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 [1].

2. Use of EAP-TNC

EAP-TNC 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 EAP-TNC does not have to do so. Therefore, EAP-TNC MUST only be used inside an EAP tunnel method that provides strong cryptographic authentication (possibly server only), message integrity and confidentiality services.

3. Definition of EAP-TNC

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

This section describes and defines the EAP-TNC method. First, it provides a protocol overview and a flow diagram. Second, it describes specific features like version negotiation and fragmentation. Third, it gives a detailed packet description. Finally, it describes how the

tls-unique channel binding [18] may be used to PA-TNC exchanges to the EAP tunnel method, defeating MITM attacks such as the Asokan attack [11].

3.1. Protocol Overview

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

The EAP-TNC method begins with the negotiation phase. The NEA Server starts this phase by sending an EAP-TNC Start message: an EAP Request message of type EAP-TNC 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 EAP-TNC 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 EAP-TNC 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 EAP-TNC method. Successful completion of EAP-TNC does not imply successful completion of the overall authentication nor does EAP-TNC 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 EAP-TNC 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

EAP-TNC version negotiation takes place in the first EAP-TNC message sent by the NEA Server (the Start message) and the first EAP-TNC 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 EAP-TNC version that the NEA Server supports and is willing to accept.

The NEA Client chooses the EAP-TNC 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 EAP-TNC versions). If the NEA Client only supports EAP-TNC versions greater than the value sent by the NEA Server, the EAP 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 EAP-TNC session immediately. Otherwise, the version sent by the NEA Client is the version of EAP-TNC 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 EAP-TNC messages in this exchange.

This specification defines version 1 of EAP-TNC. Version 0 is reserved and MUST never be sent. New versions of EAP-TNC (values 2-7) may be defined by Standards Action, as defined in RFC 5226 [6].

3.3. Fragmentation

In most cases, EAP-TNC fragmentation will not be required. But PB-TNC batches can be very long and EAP message length is sometimes tightly constrained so EAP-TNC includes a fragmentation mechanism to be used when a particular PB-TNC batch is too long to fit into a single EAP-TNC message.

The fragmentation mechanism used in EAP-TNC is quite similar to the mechanism used by EAP-TLS [17], EAP-TTLS [14], and EAP-FAST [12]. It uses the L flag (length included) and the M flag (more fragments) as well as the Data Length field.

A party (NEA Client or NEA Server) that needs to fragment a long PB-TNC batch SHOULD break the batch into pieces (called "fragments") that will fit into EAP-TNC messages. Then this party sends the fragments in proper sequence, one fragment per EAP-TNC message. The receiving party recognizes the fragments and holds them for reassembly, sending an acknowledgment for each fragment so that the next fragment can be sent (since EAP only allows one message in flight and is half duplex).

The EAP-TNC message that contains the first fragment MUST have the L flag set to indicate that fragmentation is being initiated. This packet also MUST contain the Data Length field, indicating the total octet length of the unfragmented batch and allowing the party receiving the fragments to know how much data will eventually be coming. The L flag MUST NOT be set and the Data Length field MUST NOT be present in any EAP-TNC message unless that message contains the first fragment of a fragmented PB-TNC batch. The M flag MUST be set on all but the last fragment and MUST NOT be set on the last fragment.

A party that receives an EAP-TNC message with the M flag set MUST respond with an EAP-TNC Acknowledgement message: an EAP-TNC message with no Data and with the L, M, and S flags set to 0. The party that sent an EAP-TNC message with the M flag set MUST wait for the EAP-TNC Acknowledgement packet before sending the next fragment.

EAP-TNC authenticators and NEA Clients MUST include support for EAP-TNC fragmentation with Data Lengths up to 100,000 octets. However, a NEA Server or peer still MAY decide to terminate an EAP-TNC exchange at any time for a variety of reasons.

3.4. EAP-TNC Message Format

This section provides a detailed description of the fields in an EAP-TNC message. For a description of the diagram conventions used here, see section 1.2. Since EAP-TNC 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 Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Data Length           |           Data ...            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

The Code field is one octet and identifies the type of the EAP message. The only values used for EAP-TNC 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 EAP-TNC message, starting from the Code field. If an EAP-TNC message has been fragmented, the Length field will cover only this fragment and thus doesn't reflect the overall length of the entire unfragmented EAP-TNC message.

Type

38

[IANA Note: This value was previously reserved for another purpose but has been used for EAP-TNC for some time and never used for the other purpose so please assign this value to EAP-TNC.]

Flags

```
+-+-+-+-+-+
|L M S R R|
+-+-+-+-+-+
```

L: Length included

Indicates the presence of the Data Length field in the EAP-TNC message. This flag MUST be set for an EAP-TNC message that contains the first fragment of a fragmented EAP-TNC message and only for such a message. This flag MUST NOT be set for non-fragmented messages.

M: More fragments

Indicates that more fragments are to follow. This flag MUST be set for all EAP-TNC messages that contain a fragmented EAP-TNC message except that this bit MUST NOT be set for EAP-TNC messages that contain the last fragment of a fragmented message. This flag MUST NOT be set for EAP-TNC messages that contain unfragmented Data.

S: Start

Indicates the beginning of an EAP-TNC 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 or have the L or M flags set.

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 Length

Data Length is an optional field four octets in length. It MUST be present if and only if the L flag is set. When present, it indicates the total length, before fragmentation, of a fragmented PB-TNC batch. The Data Length field MUST be set in the EAP-TNC message that contains the first in a series of fragments and MUST NOT be set in subsequent fragments.

Data

Variable length data. The length of the Data field in a particular EAP-TNC message may be determined by subtracting the length of the EAP-TNC header fields from the value of the two octet Length field. Note, however, that this data may be just one part of a longer fragmented PB-TNC batch conveyed in multiple EAP-TNC messages.

3.5. 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, the Posture Broker on an EMA-equipped endpoint SHOULD pass the tls-unique channel binding [18] for PT-EAP's tunnel method 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 tunnel. If the values match and the integrity of the endpoint is good, 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, 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 the EAP-TNC inner EAP method. As discussed throughout the document, the EAP-TNC method is designed to run inside an EAP tunnel method which is capable of protecting the EAP-TNC protocol from many threats. Since the EAP tunnel method will be specified separately, these security considerations specify requirements on the tunnel method but do not evaluate its ability to meet those requirements.

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-TNC protocol.

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

- 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

- 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 EAP-TNC is encapsulated within the protected tunnel. Therefore, this section highlights the cryptographic requirements that need to be met by the EAP tunnel method carrying EAP-TNC 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 EAP-TNC 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 EAP-TNC 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 again NEA assessment message theft, the EAP tunnel method carrying EAP-TNC MUST provide strong cryptographic authentication, integrity and confidentiality protection. The use of bi-directional authentication in the EAP tunnel method carrying EAP-TNC ensures that only properly authenticated and authorized parties may be involved in an assessment message exchange. When EAP-TNC is carried within a cryptographically protected EAP tunnel method like 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 EAP-TNC 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. For example, the EAP-TTLS method's use of hashing algorithms provides strong integrity protections that allow for detection of any changes in the

content of the message exchange. Additionally, adversaries are unable to observe the EAP-TNC method housed inside of an encrypting EAP tunnel method (e.g. EAP-TTLS) because the messages are encrypted by the TLS [2] ciphers, so an attacker would have difficulty in determining where to insert the falsified message, since the attacker is unable to determine where the message boundaries exist.

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 EAP-TNC 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 EAP-TNC message recipients to detect message alterations by other types of network based adversaries. Because EAP-TNC does not itself provide explicit integrity protection for the EAP-TNC 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 EAP-TNC 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 3.5. and 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. Section 3.5. 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. 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.

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 includes the following requirement that is to be applied during the selection of the EAP tunnel method(s) used in conjunction with EAP-TNC:

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. TLS server when EAP-TTLS is used) 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 EAP-TNC.

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 both or one party performing a cryptographic 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 Client, so the session is cryptographically bound to a particular authentication event.

To support countermeasures against NEA Asokan attacks as described in section 3.5. the EAP Tunnel Method used with EAP-TNC 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.

4.4. Candidate EAP Tunnel Method Protections

This section discusses how EAP-TNC is used within various EAP tunnel methods to meet the PT requirements from section 4.3.

EAP-FAST and EAP-TTLS make use of TLS [2] 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. EAP-TNC 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, 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 certificates. 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 EAP-TNC.) 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.

4.5. Security Claims for EAP-TNC as per RFC3748

This section summarizes the security claims 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:	Yes
Channel binding:	No

5. 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 EAP-TNC inner method. Therefore, the Posture Transport Client and Posture Transport Server's implementation of EAP-TNC MUST NOT observe the contents of the carried PB-TNC batches that could contain PII carried by PA-TNC or PB-TNC.

While EAP-TNC 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 EAP-TNC payload.

6. IANA Considerations

This document defines an EAP method type named EAP-TNC with the value 38.

[IANA Note: This value was previously reserved for another purpose but has been used for EAP-TNC for some time and never used for another purpose so please assign this value to EAP-TNC.]

This document also defines one new IANA registry: EAP-TNC 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).

6.1. Registry for EAP-TNC Versions

The name for this registry is "EAP-TNC 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 EAP-TNC Versions. Additional entries to this registry are added by Standards Action, as defined in RFC 5226 [6].

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

7. References

7.1. Normative References

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

[2] Dierks T., Rescorla E., "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

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

[4] Sahita, R., Hanna, S., and R. Hurst, "PB-TNC: A Posture Broker Protocol (PB) Compatible with TNC", RFC 5793, March 2010.

[5] LAN/MAN Standards Committee of the IEEE Computer Society, Standard for Local and Metropolitan Area Networks - Port Based Network Access Control, IEEE Std. 802.1X-2004, December 2004.

[6] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008.

7.2. Informative References

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

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

[9] Sangster, P., "PT-TLS: A Posture Transport Protocol (PT) Compatible with TNC Using Transport Layer Security (TLS)", draft-sangster-nea-pt-tls-02.txt, work in progress, March 2011.

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

[11] N. Asokan, Valtteri Niemi, Kaisa Nyberg, "Man in the Middle Attacks in Tunneled Authentication Protocols", Nokia Research Center, Finland, Nov. 11, 2002, http://eprint.iacr.org/2002/163.pdf

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

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

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

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

[16] J. Salowey, S. Hanna, "NEA Asokan Attack Analysis", draft-salowey-nea-asokan-00.txt (work in progress), October 2010.

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

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

8. Acknowledgments

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, 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, Paul Sangster, Dean Sheffield, Curtis Simonson, Jeff Six, Ned Smith,

Michelle Sommerstad, Joseph Tardo, Lee Terrell, Chris Trytten, and John Vollbrecht.

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A. Evaluation Against NEA Requirements

This section evaluates the PT-EAP protocol against the PT requirements defined in the NEA Overview and Requirements and PB-TNC specifications. Each subsection considers a separate requirement and highlights how PT-EAP meets the requirement.

A.1. Evaluation Against Requirement C-1

Requirement C-1 says:

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

PT-EAP meets this requirement. Use of the EAP protocol along with EAP-TNC and suitable EAP tunnel methods will allow for multiple roundtrips.

A.2. Evaluation Against Requirements C-2

Requirement C-2 says:

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-EAP does NOT meet this requirement. Generally EAP is used by the endpoint during the joining of the network. At that time, the endpoint lacks an IP address so is unable to accept inbound posture assessment requests from the NEA Server. Subsequent reassessments of the endpoint after it has been given access to a portion of the IP network can use the PT-TLS protocol that supports the NEA Client and NEA Server to initiate an assessment.

A.3. Evaluation Against Requirements C-3

Requirement C-3 says:

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-EAP meets this requirement by leveraging the security capabilities of the underlying EAP tunnel method. EAP-TNC itself does not provide protection against a variety of

potential attacksso it relies on cryptographic support by the EAP tunnel method.

A.4. Evaluation Against Requirements C-4

Requirement C-4 says:

C-4 The PA and PB protocols MUST be capable of operating over any PT protocol. For example, the PB protocol must provide a transport independent interface allowing the PA protocol to operate without change across a variety of network protocol environments (e.g. EAP/802.1X, PANA, TLS and IKE/IPsec).

Not applicable to PT, but PT-EAP is independent of PA and PB allowing those protocols to operate over other PT protocols.

A.5. Evaluation Against Requirements C-5

Requirement C-5 says:

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.

Based on this requirement, PT-EAP should receive a strong preference. PT-EAP is compatible with IF-T Binding to Tunneled EAP Methods 1.1, an open TCG specification that has been widely implemented.

A.6. Evaluation Against Requirements C-6

Requirement C-6 says:

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-EAP meets this requirement. The PT-EAP protocol is independent of the number of Posture Collectors and Posture Validators.

A.7. Evaluation Against Requirements C-7

Requirement C-7 says:

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

PT-EAP meets this requirement, subject to the limitations of the underlying EAP protocol. PT-EAP allows for the transport of a very large number of attributes, up to 2^32 - 1 octets per PB-TNC batch. Furthermore, the PT-EAP protocol transports data efficiently, only adding 10 octets of overhead per PT-EAP message, which is small considering that a single PT-EAP message may carry multiple PA-TNC attributes.

However, it is important to note that the EAP protocol that underlies PT-EAP is not a good choice for transporting large amounts of data. EAP only supports one packet in flight at a time, which severely limits throughput. Further, some network equipment imposes timeout restrictions on EAP exchanges. Therefore, PT-EAP should not be used to transport large amounts of attributes.

A.8. Evaluation Against Requirements C-8

Requirement C-8 says:

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

PT-EAP protocols meet this requirement. PT-EAP was designed to minimize the amount of overhead included in the protocol to allow for efficient use over bandwidth or latency constrained network links.

A.9. Evaluation Against Requirements C-9

Requirement C-9 says:

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-EAP meets this requirement. PT-EAP does not include messages intended for display to the user.

A.10. Evaluation Against Requirements C-10

Requirement C-10 says:

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

PT-EAP meets this requirement. The PT-EAP protocol does not include any strings in its fields but it allows higher-layer protocols to encode their strings in UTF-8 format. This allows the protocol to support a wide range of languages efficiently.

A.11. Evaluation Against Requirements C-11

Requirement C-11 says:

C-11 Due to the potentially different transport characteristics provided by the underlying candidate PT protocols, the NEA Client and NEA Server MUST be capable of becoming aware of and adapting to the limitations of the available PT protocol. For example, some PT protocol characteristics that might impact the operation of PA and PB include restrictions on: which end can initiate a NEA connection, maximum data size in a message or full assessment, upper bound on number of roundtrips, and ordering (duplex) of messages exchanged. The selection process for the PT protocols MUST consider the limitations the candidate PT protocol would impose upon the PA and PB protocols.

PT-EAP meets this requirement. The PT-EAP implementations may be limited in number of roundtrips, assessment overall time, or data transmission. These constraints will be exposed up the protocol stack so the Posture Broker Client and Posture Broker Server can optimize and make most efficient use of the available resources during the assessment.

A.12. Evaluation Against Requirements PT-1

Requirement PT-1 says:

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-EAP meets this requirement. The PT-EAP encapsulates PB-TNC batches without interpreting their contents.

A.13. Evaluation Against Requirements PT-2

Requirement PT-2 says:

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-EAP meets this requirement. The PT-EAP protocol leverages an EAP tunnel method to provide mutual authentication, integrity protection and confidentiality as well as replay protection. For more information see the Security Considerations in section 4.

A.14. Evaluation Against Requirements PT-3

Requirement PT-3 says:

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.

EAP-TNC includes support for fragmentation and the underlying EAP tunnel methods include support for duplicate detection and reordering to provide in-sequence delivery.

A.15. Evaluation Against Requirements PT-4

Requirement PT-4 says:

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

PT-EAP meets this requirement. The PT-EAP operates on top of the 802.1X and IKEv2 protocols.

A.16. Evaluation Against Requirements PT-5

Requirement PT-5 says:

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-EAP does NOT meet this requirement. PT-EAP is intended for a different usage. PT-EAP is intended to be used for pre-network admission before the endpoint has been given an IP address and routes on the network. This means that network layer protocols such as IP are not yet able to communicate with

the system. The PT-TLS (PT Binding to TLS) [9] meets this requirement.

A.17. Evaluation Against Requirements PT-6 (from PB-TNC specification)

Requirement PT-6 says:

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

PT-EAP meets this requirement. The PT-EAP fits into the EAP framework which provides for orderly initiation and shutdown.

A.18. Evaluation Against Requirements PT-7 (from PB-TNC specification)

Requirement PT-7 says:

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

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

A.19. Evaluation Against Requirements PT-8 (from PB-TNC specification)

Requirement PT-8 says:

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

PT-EAP meets this requirement. The PT-EAP utilizes EAP's half duplex, round robin message exchange to provide flow and congestion control.

A.20. Evaluation Against Requirements PT-9 (from PB-TNC specification)

Requirement PT-9 says:

PT-9 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-EAP specification meets this requirement. This specification discusses the level of transport service provided to the Posture Broker Client and Posture Broker Server. Generally, the PT-EAP method supports the pre-network admission usages discussed in RFC 5209. The maximum message size for PT-EAP is 2^16-10 octets. EAP by its nature is half duplex and

simple which allows it to be used in a wide variety of settings including over link layer protocols during the entrance to the network.

Authors' Addresses

Steve Hanna
Juniper Networks, Inc.
79 Parsons Street
Brighton, MA 02135 USA
Email: shanna@juniper.net

Paul Sangster
Symantec Corporation
6825 Citrine Drive
Carlsbad, CA 92009 USA
Email: paul_sangster@symantec.com