Internet Engineering Task Force (IETF)
Request for Comments: 5792
Category: Standards Track
ISSN: 2070-1721

P. Sangster
Symantec Corporation
K. Narayan
Cisco Systems

March 2010

PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)

Abstract

This document specifies PA-TNC, a Posture Attribute protocol identical to the Trusted Computing Group's IF-M 1.0 protocol. The document then evaluates PA-TNC against the requirements defined in the NEA Requirements specification.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5792.

Table of Contents

1. Introduction

This document specifies PA-TNC, a Posture Attribute (PA) Protocol identical to the Trusted Computing Group's IF-M 1.0 protocol [8]. The document then evaluates PA-TNC against the requirements defined in the Network Endpoint Assessment (NEA) Requirements specification [9].

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 Overview and Requirements specification. The reader is assumed to be thoroughly familiar with that document. No familiarity with TCG specifications is assumed.

1.2. Message Diagram Conventions

This specification defines the syntax of PA-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 1-octet field with only bit 0 set has the value 0x80.

1.3. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

2. Design Considerations

This section discusses some of the key design considerations for the PA protocol.

2.1. Standard Attribute Namespace for Interoperability

The PA protocol requires the use of two categories of namespaces: component types (AKA PA subtypes) and attributes. Each of these namespace categories needs to contain well-known, interoperable names with defined syntax and semantics co-existing with names for vendor-

defined private extensions. Similarly, each namespace category needs to be readily extensible without repeated coordination yet avoids naming conflicts.

The PA-TNC and PB-TNC protocols provide for multiple orthogonal namespaces for each category that exist without overlap by including a Structure of Management Information (SMI) Private Enterprise Number (PEN) field to identify the definer of namespace of the associated field. This allows the IETF NEA WG to define a set of standard component types and attribute types while allowing vendors to each create additional names outside of the IETF standard namespace. Over time, vendor-defined names might be proposed for standardization and thus migration into the IETF namespace.

The PB-TNC protocol defines an IETF standard namespace (using vendor-id=0) that allows for definition of standard component types (e.g., Operating System, Firewall, Anti-Virus) using the PA Subtype field (see section 3.2). Similarly, PA-TNC defines a set of standard attributes in section 4.2 that represent the most common capabilities (attributes) of these types of components across a variety of vendor implementations. The standard namespace allows NEA deployments with both open source and vendor-provided NEA implementations to support a consistent set of policies across their environment based on these standard attributes. The standard attributes can be used with a variety of endpoints (hosts, printers, mobile devices) that are running applications and operating systems (defined by the PA subtypes) from a variety of vendors.

2.2. Vendor-Defined Namespace for Differentiation and Agility

The endpoint is a very dynamic environment in terms of rate of new features being deployed and attacks that are crafted against existing and new applications such as viruses, worms, malware, and spyware. It is difficult to imagine the standard namespaces being able to keep pace with this rapidly changing environment. Vendors typically differentiate themselves by moving rapidly to provide unique mechanisms to address such threats and their ability to deal with changes in an agile manner. The PA-TNC and PB-TNC protocols allow for creation of vendor-defined namespace(s) where each namespace allows use of vendor-defined PA subtypes to identify non-standard applications or operating system variants and vendor-defined attributes describing new aspects of each type of component. The vendor namespaces will allow NEA deployments to craft compliance policies using a mixture of attributes from both the IETF standard namespace and vendor-defined namespaces that may include multiple vendors representing the various hardware and software components present on the endpoints.

The PA-TNC protocol's use of vendor-id to identify the namespace of each attribute allows Posture Collectors to support some or all of the IETF standard attributes plus optionally a set of vendor-defined attributes (potentially from more than one vendor-id namespace). For instance, an open source anti-virus Posture Collector might be written that supports all of the IETF standard attributes used to describe a local anti-virus component and a subset of multiple anti-virus manufacturers' vendor-defined attributes. This Posture Collector might therefore be able to interoperate with Posture Validators from multiple vendors. Conversely, a simple Posture Collector might be written to ignore any vendor-defined attributes requested and only return standard attributes that it supports. If the vendor-provided Posture Validator's policy allows for this subset to be considered compliant, then these simple Posture Collectors can be used to perform a successful assessment.

2.3. Use of TLV-Based Encoding for Efficiency

The PA-TNC protocol has chosen to employ a binary encoding using a type-length-value (TLV) structure. TLV encoding was preferred over the use of a textual encoding format such as XML to provide a more efficient utilization of the potentially constrained bandwidth available between the NEA Client and NEA Server (see NEA Overview and Architecture [9]). Efficiency was a primary criterion for this choice with consideration given to both:

1. Optimization of the bits-on-the-wire to accommodate NEA requirements for assessment over low bandwidth or high latency links (C-8) and allow for the Posture Transport (PT) protocol to run over existing network access protocols (PT-4, C-11) that are constrained by packet size.

2. Optimization of CPU utilization on the endpoint to accommodate for low power endpoints such as mobile devices.

The choice of TLV encoding does not preclude the use of XML-based attribute values within the vendor namespaces or future standard attributes. It is conceivable that certain vendors may utilize XML encoding for extensibility within their namespace when the above considerations are less applicable to their technologies. Attributes encoded within the vendor-defined namespace using alternate encoding such as XML will be opaque to NEA software only supporting standard attributes and will be processed primarily by the vendor-defined components (collector/validator).

3. PA-TNC Message Protocol

This section discusses the use of the PA-TNC message and its attributes, and specifies the syntax and semantics for the PA-TNC message header. The details of each attribute included within the PA-TNC payload are specified in section 4.2.

3.1. PA-TNC Messaging Model

PA-TNC messages are carried by the PB-TNC protocol [5], which provides a multi-roundtrip reliable transport and end-to-end message delivery to subscribed (interested) parties using a variety of underlying network protocols. PA-TNC is unaware of these underlying PT protocols being used below PB-TNC.

The interested parties consist of Posture Collectors on the NEA Client and Posture Validators associated with the NEA Server that have registered to receive messages about particular types of components (e.g., anti-virus) during an assessment. The PA-TNC messaging protocol operates synchronously within an assessment session, with Posture Collectors and Posture Validators taking turns sending one or more messages to each other. Each PA-TNC message may contain one or more attributes associated with the functional component identified in the component type (PA Subtype) of the Posture Broker (PB) protocol.

Posture Collectors may only send PA-TNC messages to Posture Validators and vice versa. No Posture Collector-to-Posture Collector or Posture Validator-to-Posture Validator messaging is allowed to occur. Each Posture Collector or Posture Validator may send several PA-TNC messages in succession before indicating that it has completed its batch of messages to the Posture Broker Client or Posture Broker Server respectively. As necessary, the Posture Broker Client and Posture Broker Server will batch these messages prior to sending them over the network.

PB-TNC provides a publish/subscribe model of message exchange. This means that, at any given point in time, zero or more subscribers for a particular type of message may be present on a Posture Broker Client or Posture Broker Server. This is beneficial, since it allows one Posture Collector or Posture Validator to combine multiple functions (like anti-virus and personal firewall) by subscribing to both TNC standard component types. It also allows multiple Posture Collectors or Posture Validators to support the same components, such as two anti-virus Posture Validators that are each used to manage their own respective anti-virus client software.

However, this publish/subscribe model has some possible negative side effects. When a Posture Collector or Posture Validator initially sends a PA-TNC message, it does not know whether it will receive many, one, or no PA-TNC messages from the other side. For many types of assessments, this is acceptable, but in some cases a more direct channel binding between a particular Posture Collector and Posture Validator pair is necessary. For example, a Posture Validator may wish to provide remediation instructions to a particular Posture Collector that it knows is capable of remediating a non-compliant component. This can be accomplished using the exclusive delivery PB-TNC capability to limit distribution of a message to a single Posture Collector by including the target Posture Collector Identifier in the PB-PA header. For more information on the PB-PA header, see section 4.5 of the PB-TNC specification.

3.2. PA-TNC Relationship to PB-TNC

This section summarizes the major elements of a PA-TNC message as they might appear inside of a PB-TNC message. The double line (===) in the diagram below indicates the separation between the PB-TNC and PA-TNC protocols. The PA-TNC portion of the message is delivered to each Posture Collector or Posture Validator registered to receive messages containing a particular message type. Note that PB-TNC is capable of carrying multiple PB-TNC and PA-TNC messages in a single PB-TNC batch. See the PB-TNC specification [5] for more information on its capabilities.

One important linkage between the PA-TNC and PB-TNC protocols is the PA message type (PA Message Vendor ID and PA Subtype) that is used by the Posture Broker Client and Posture Broker Server to route messages to interested Posture Collectors and Posture Validators. The message type indicates the software component (component type) that is associated with the attributes included inside the PA-TNC message. Therefore, Posture Collectors and Posture Validators written to support an assessment of a particular component can register to receive messages about the component and thus participate in its assessment. Each Posture Collector and Posture Validator MUST only send PA-TNC messages containing attributes that pertain to the software component defined in the message type of the message. This ensures that only the appropriate Posture Collectors and Posture Validators that support a particular type of component will receive attributes related to that component. If a PA-TNC message contained a mix of attributes about different components and a message type of only one of those components, the message would only be delivered to parties interested in the component type included in the message type, so other interested recipients wouldn’t see those attributes.

The message type is composed of two fields: a PA Message Vendor ID and a PA Subtype. The PA Message Vendor ID identifies the vendor or other organization that defined this message type. The PA Subtype identifies the message type more specifically within the set of message types defined by that vendor. This specification defines several IETF Standard PA Subtypes to be used with a PA Message Vendor ID of zero (0). Within this specification, the PA Subtype field is used to indicate the type of component (e.g., firewall) involved with the message's attributes. Therefore, for clarity, the PA subtype will be referred to as the "component type" in this specification. Vendor-defined namespaces may use other semantics for the PA Subtype field as this is outside the scope of this specification.

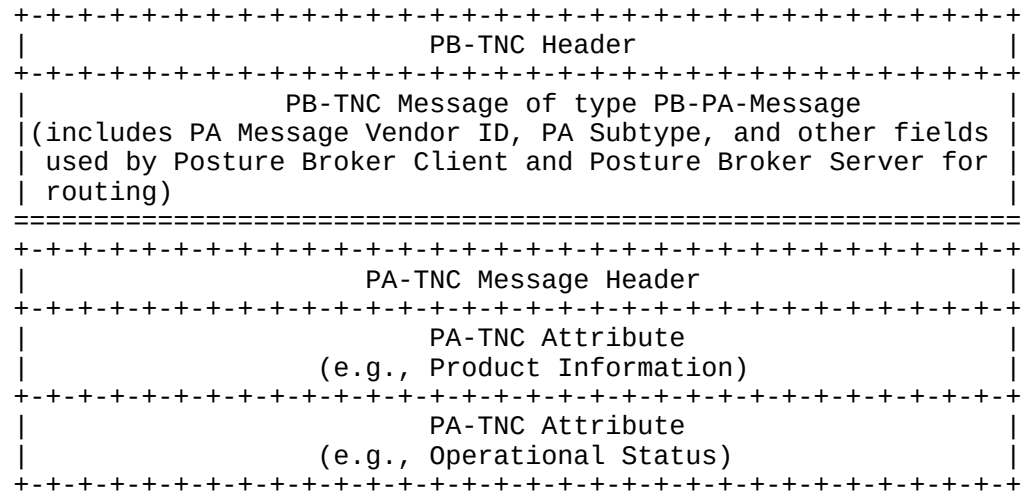


Figure 1. Overview of a PB-TNC batch that contains a PA-TNC message

For example, if a Posture Broker Client sent a PB-TNC batch that contained a PA-TNC message with a message type indicating firewall component, this message would be routed by the Posture Broker Server to Posture Validators registered to assess firewalls. Each registered Posture Validator would receive a copy of the PA-TNC message including the PA-TNC header and set of attributes. It is important that each of the attributes included in the PA-TNC message be associated with the firewall component because only the Posture Collector and Posture Validator interested in firewalls will receive such messages.

If the above message contained both firewall and operating system attributes inside a PA-TNC message with a component type of firewall, then any Posture Collector and Posture Validator registered to receive operating system messages would not receive those attributes, as the messages would only be delivered to those registered for firewall messages.

3.3. PB-PA Posture Collector and Posture Validator Identifiers

The PB-PA header contains several fields important to the processing of a received PA message. The PA Vendor ID and Subtype are described in the PB-TNC specification and above in section 3.2. Also present in the PB-PA header is a pair of fields that identify the Posture Collector and/or Posture Validator involved in the exchange. These fields are used for performing exclusive delivery of messages as described in section 3.1 and as an indicator for correlation of received attributes.

Correlation of attributes is necessary when the sending Posture Collector provides posture for multiple implementations of a single type of component during an assessment, so the recipient Posture Validators need to know which attributes are describing the same implementation.

For example, a single Posture Collector might report attributes on two installed VPN implementations on the endpoint. Because the individual attributes do not include an indication of which VPN product they are describing, the recipient needs something to perform this correlation. Therefore, for this example, the VPN Posture Collector would need to obtain two Posture Collector Identifiers from the Posture Broker Client and consistently use one with each of the implementations during an assessment. The VPN Posture Collector would group all the attributes associated with a particular VPN implementation into a single PB-PA message and send the message using the Posture Collector Identifier it designates as going with the particular implementation. This approach allows the recipient to recognize when attributes in future assessment messages also describe the same component implementation.

3.4. PA-TNC Messages in PB-TNC

As depicted in section 3.2, a PA-TNC message consists of a PA-TNC header followed by a sequence of one or more attributes. The PA-TNC message header (described in section 3.6) and the header for each of the PA-TNC attributes (specified in section 4.1) have a fixed type-length-value (TLV) format. Each PA-TNC message MAY contain a mixture of standards-based and vendor-defined attributes identifiable using the type portion of the attribute header. All Posture Collectors and

Posture Validators compliant with this specification MUST be capable of processing multiple attributes in a received PA-TNC message. A Posture Collector or Posture Validator that receives a PA-TNC message can use the attribute header's length field to skip any attributes that it does not understand, unless the attribute is marked as mandatory to process.

3.5. IETF Standard PA Subtypes

This section defines several IETF Standard PA Subtypes. Each PA subtype defined here identifies a specific component relevant to the endpoint's posture. This allows a small set of generic PA-TNC attributes (e.g., Product Information) to be used to describe a large number of different components (e.g., operating system, anti-virus, etc.). It also allows Posture Collectors and Posture Validators to specialize in a particular component and only receive PA-TNC messages relevant to that component.

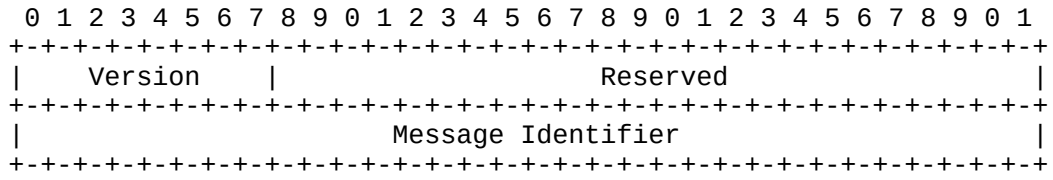
Value	Integer	Definition
0	Testing	Reserved for use in specification examples, experimentation and testing.
1	Operating System	Operating system running on the endpoint
2	Anti-Virus	Host-based anti-virus software
3	Anti-Spyware	Host-based anti-spyware software
4	Anti-Malware	Host-based anti-malware (e.g., anti-bot) software not included within anti-virus or anti-spyware components
5	Firewall	Host-based firewall
6	IDPS	Host-based Intrusion Detection and/or Prevention Software (IDPS)
7	VPN	Host-based Virtual Private Network (VPN) software
8	NEA Client	NEA client software

These PA subtypes must be used in a PB-PA message with a PA Message Vendor ID of zero (0) indicating an IETF standard type of component (as described in the PB-TNC specification [5]). If these PA subtype

values are used with a different PA Message Vendor ID, they have a completely different meaning that is not defined in this specification. Posture Collectors and Posture Validators MUST NOT require support for particular vendor-specific PA subtypes and MUST interoperate with other parties despite any differences in the set of vendor-specific PA subtypes supported (although they MAY permit administrators to configure them to require support for specific PA subtypes).

3.6. PA-TNC Message Header Format

This section describes the format and semantics of the PA-TNC header. Every PA-TNC message MUST start with a PA-TNC header. The PA-TNC header provides a common context applying to all of the attributes contained within the PA-TNC payload. The payload consists of a sequence of assessment attributes described in section 4.2.



Version

This field indicates the version of the format for the PA-TNC message. This version is intended to allow for evolution of the PA-TNC message header and payload in a manner that can easily be detected by message recipients.

PA-TNC message senders MUST set this field to 0x01 for all PA-TNC messages that comply with this specification. Implementations responding to a PA-TNC message containing a supported version MUST use the same version number to minimize the risk of version incompatibility. Message recipients MUST respond to a PA-TNC message containing an unsupported version by sending a Version Not Supported error in a PA-TNC Error attribute that is the only PA-TNC attribute in a PA-TNC message with version number 1.

PA-TNC message initiators supporting multiple PA-TNC protocol versions SHOULD be able to alter which version of PA-TNC message they send based on prior message exchanges with a particular peer Posture Collector or Posture Validator.

Reserved

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

Message Identifier

This field contains a value that uniquely identifies this message, differentiating it from others sent by a particular PA-TNC message sender within this assessment. This value can be included in the payload of a response message to indicate which message was received and caused the response. This value is included in the payload of PA-TNC error messages so the party who receives the error message can determine which of the messages they had sent caused the error.

PA-TNC message senders MUST NOT send the same message identifier more than once during an assessment. Message identifiers may be randomly generated or sequenced as long as values are not repeated during an assessment message exchange. PA-TNC message recipients are not required to check for duplicate message identifiers.

4. PA-TNC Attributes

This section defines the PA-TNC attributes that can be carried within a PA-TNC message. The initial section defines the standard attribute header that appears at the start of each attribute in a PA-TNC message. The second section defines each of the IETF Standard PA-TNC Attributes and the final section discusses how vendor-defined PA-TNC attributes can be used within a PA-TNC message. Vendor-defined PA-TNC attributes use the vendor's SMI Private Enterprise Number in the Attribute Type field.

A PA-TNC message MUST contain a PA-TNC header (defined in section 3.6. followed by a sequence of zero or more PA-TNC attributes. All PA-TNC attributes MUST begin with a standard PA-TNC attribute header, as defined in section 4.1. The contents of PA-TNC attributes vary widely, depending on their attribute type. Section 4.2 defines the IETF Standard PA-TNC Attributes. Section 4.3 discusses how vendor-specific PA-TNC attributes can be defined.

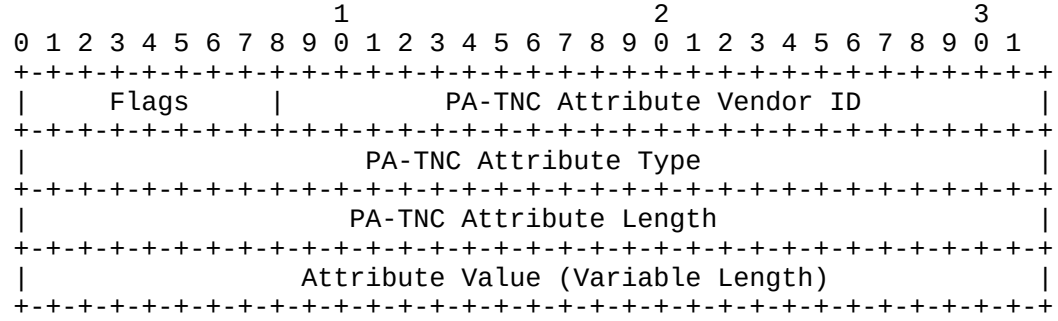
4.1. PA-TNC Attribute Header

Following the PA-TNC message header is a sequence of zero or more attributes. All PA-TNC attributes MUST begin with the standard PA-TNC attribute header defined in this subsection. Each attribute described in this specification is represented by a TLV tuple. The TLV tuple includes an attribute identifier comprised of the Vendor ID

and Attribute Type (type), the TLV tuple's overall length, and finally the attribute's value. The use of TLV representation was chosen due to its flexibility and extensibility and use in other standards. Recipients of an attribute can use the attribute type fields to determine the precise syntax and semantics of the attribute value field and the length to skip over an unrecognized attribute. The length field is also beneficial when a variable-length attribute value is provided.

The TLV format does not contain an explicit TLV format version number, so every attribute included in a particular PA-TNC message MUST use the same TLV format. Using the PA-TNC message version number to indicate the format of all TLV attributes within a PA-TNC message allows for future versioning of the TLV format in a manner detectable by PA-TNC message recipients. Similarly, requiring all TLV attribute formats to be the same within a PA-TNC message also ensures that recipients compliant with a particular PA-TNC message version can at least parse every attribute header and use the length to skip over unrecognized attributes. Finally, all attribute TLVs within a PA-TNC message MUST pertain to the same implementation of the component. This restriction is relevant when a single Posture Collector is reporting on multiple implementations of a component, so must send multiple PA-TNC messages each including only the attributes describing a single implementation. For more information on how Posture Collectors should handle multiple implementations, see section 3.3.

Every PA-TNC-compliant TLV attribute MUST use the following TLV format:



Flags

This field defines flags impacting the processing of the associated attribute.

Bit 0 (0x80) is the NOSKIP flag. Any Posture Collector or Posture Validator that receives an attribute with this flag set to 1 but does not support this attribute MUST NOT process any part of the PA-TNC message and SHOULD respond with an Attribute Type Not Supported error in a PA-TNC error message.

In order to avoid taking action on a subset of the attributes only to later find an unsupported attribute with the NOSKIP flag set, recipients of a multi-attribute PA-TNC message might need to scan all of the attributes prior to acting upon any attribute.

When the NOSKIP flag is set to 0, recipients SHOULD skip any unsupported attributes and continue processing the next attribute.

Bit 1-7 are reserved for future use. These bits MUST be set to 0 on transmission and ignored upon reception.

PA-TNC Attribute Vendor ID

This field indicates the owner of the namespace associated with the PA-TNC Attribute Type. This is accomplished by specifying the 24-bit SMI Private Enterprise Number Vendor ID of the party who owns the Attribute Type namespace. IETF Standard PA-TNC Attribute Types MUST use zero (0) in this field.

The PA-TNC Attribute Vendor ID 0xffffff is reserved. Posture Collectors and Posture Validators MUST NOT send PA-TNC messages in which the PA-TNC Attribute Vendor ID has this reserved value (0xffffff). If a Posture Collector or Posture Validator receives a message in which the PA-TNC Attribute Vendor ID has this reserved value (0xffffff), it SHOULD respond with an Invalid Parameter error code in a PA-TNC Error attribute.

PA-TNC Attribute Type

This field defines the type of the attribute included in the Attribute Value field. This field is qualified by the PA-TNC Attribute Vendor ID field so that a particular PA-TNC Attribute Type value (e.g., 327) has a completely different meaning depending on the value in the PA-TNC Attribute Vendor ID field. Posture Collectors and Posture Validators MUST NOT require support for particular vendor-specific PA-TNC Attribute Types and MUST interoperate with other parties despite any differences in the set of vendor-specific PA-TNC Attribute Types supported (although they MAY permit administrators to configure them to require support for specific PA-TNC attribute types).

If the PA-TNC Attribute Vendor ID field has the value zero (0), then the PA-TNC Attribute Type field contains an IETF Standard PA-TNC Attribute Type, as listed in the IANA registry. IANA maintains a registry of PA-TNC Attribute Types. Entries in this registry are added by Expert Review with Specification Required, following the guidelines in section 7. Section 4.2 of this specification defines the initial set of IETF Standard PA-TNC Attribute Types.

The PA-TNC Attribute Type 0xffffffff is reserved. Posture Collectors and Posture Validators MUST NOT send PA-TNC messages in which the PA-TNC Attribute Type has this reserved value (0xffffffff). If a Posture Collector or Posture Validator receives a message in which the PA-TNC Attribute Type has this reserved value (0xffffffff), it SHOULD respond with an Invalid Parameter error code in a PA-TNC Error attribute.

PA-TNC Attribute Length

This field contains the length in octets of the entire PA-TNC attribute including the PA-TNC Attribute Header (the fields Flags, PA-TNC Attribute Vendor ID, PA-TNC Attribute Type, and PA-TNC Attribute Length). Therefore, this value MUST always be at least 12. Any Posture Collector or Posture Validator that receives a message with a PA-TNC Attribute Length field whose value is less than 12 SHOULD respond with an Invalid Parameter PA-TNC error code. Similarly, if a Posture Collector or Posture Validator receives a PA-TNC message for an Attribute Type that has a well-known Attribute Value length (e.g., fixed-length attribute value) and the Attribute Length indicates a different value (greater or less than the expected value), the recipient SHOULD respond with an Invalid Parameter PA-TNC error code.

Implementations that do not support the specified PA-TNC Attribute Type can use this length to skip over this attribute to the next attribute. Note that while this field is 4 octets the maximum usable attribute length is less than 2^32-1 due to limitations of the underlying protocol stack. Specifically, PB-TNC TLV header's Batch Length field is also 32 bits in length. Therefore, the maximum batch that PB-TNC can carry is 2^32-1, so the largest PA-TNC message carried by PB-TNC must be less than 2^32-1 - size of the PB-TNC header (see section 4.1 of PB-TNC for more details).

Attribute Value

This field varies depending on the particular type of attribute being expressed. The contents of this field for each of the IETF Standard PA-TNC Attribute Types are defined in section 4.2.

4.2. IETF Standard PA-TNC Attribute Types

This section defines an initial set of IETF Standard PA-TNC Attribute Types. These Attribute Types MUST always be used with a PA-TNC Vendor ID of zero (0). If these PA-TNC Attribute Type values are used with a different PA-TNC Vendor ID, they have a completely different meaning that is not defined in this specification.

The following table briefly describes each attribute and defines the numeric value to be used in the PA-TNC Attribute Type field of the PA-TNC Attribute Header. Later subsections provide detailed specifications for each PA-TNC Attribute Value.

Number	Integer	Description
0	Testing	Reserved for use in specification examples, experimentation, and testing.
1	Attribute Request	Contains a list of attribute type values defining the attributes desired from the Posture Collectors.
2	Product Information	Manufacturer and product information for the component.
3	Numeric Version	Numeric version of the component.
4	String Version	String version of the component.
5	Operational Status	Describes whether the component is running on the endpoint.
6	Port Filter	Lists the set of ports (e.g., TCP port 80 for HTTP) that are allowed or blocked on the endpoint.
7	Installed Packages	List of software packages installed on endpoint that provide the requested component.
8	PA-TNC Error	PA-TNC message or attribute processing error.

9	Assessment Result	Result of the assessment performed by a Posture Validator.
10	Remediation Instructions	Instructions for remediation generated by a Posture Validator.
11	Forwarding Enabled	Indicates whether packet forwarding has been enabled between different interfaces on the endpoint.
12	Factory Default Password Enabled	Indicates whether the endpoint has a factory default password enabled.

The following subsections discuss the usage, format, and semantics of the Attribute Value field for each IETF Standard PA-TNC Attribute Type.

4.2.1. Attribute Request

This PA-TNC Attribute Type allows a Posture Validator to request certain attributes from the registered set of Posture Collectors.

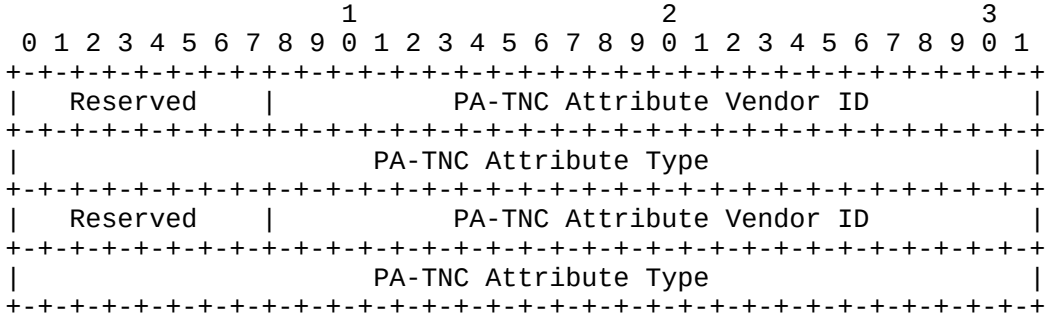
All Posture Collectors that implement any of the IETF Standard PA Subtypes defined in this specification SHOULD support receiving and processing this attribute type for at least those PA subtypes. This requirement is only a "should" because there are deployment scenarios (e.g., see section A.1) where the Posture Collectors proactively send a set of attributes at the start of an assessment (e.g., based upon local policy), so does not need to support Posture Validator requested attributes. Posture Collectors that receive but do not support the Attribute Request attribute MUST respond with an Attribute Type Not Supported PA-TNC error code. Posture Collectors that receive and process this attribute MAY choose to send all, a subset, or none of the requested attributes but MUST NOT send attributes that were not requested (except Error attributes). All Posture Validators that implement any of the IETF Standard PA Subtypes defined in this specification SHOULD support sending this attribute type for at least those PA subtypes.

Posture Validators MUST NOT include this attribute type in an Attribute Request attribute. It does not make sense for a Posture Validator to request that a Posture Collector send an Attribute Request attribute.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 1.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.

Note that this diagram shows two attribute types. The actual number of attribute types included in an Attribute Request attribute can vary from one to a large number (limited only by the maximum message and length supported by the underlying PT protocol). However, each Attribute Request MUST contain at least one attribute type. Because the length of a PA-TNC Attribute Vendor ID paired with a PA-TNC Attribute Type and a 1-octet Reserved field is always 8 octets, the number of requested attributes can be easily computed using the PA-TNC Attribute Length field by subtracting the number of octets in the PA-TNC Attribute Header and dividing by 8. If the PA-TNC Attribute Length field is invalid, Posture Collectors SHOULD respond with an Invalid Parameter PA-TNC error code.



Reserved

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

PA-TNC Attribute Vendor ID

This field contains the SMI Private Enterprise Number of the organization that controls the namespace for the following PA-TNC Attribute Type. This field enables IETF Standard PA-TNC Attributes and vendor-defined PA-TNC attributes to be used without potential collisions.

Any IETF Standard PA-TNC Attribute Types defined in section 4.2 MUST use zero (0) in this field. Vendor-defined attributes MUST use the SMI Private Enterprise Number of the organization that defined the attribute.

PA-TNC Attribute Type

The PA-TNC Attribute Type field (together with the PA-TNC Vendor ID field) indicates the specific attribute requested. Some IETF Standard PA-TNC Attribute Types MUST NOT be requested using this field (e.g., requesting a PA-TNC Error attribute). This is explicitly indicated in the description of those PA-TNC Attribute Types. Any Posture Collector or Posture Validator that receives an Attribute Request containing one of the prohibited Attribute Types SHOULD respond with an Invalid Parameter error in a PA-TNC error message.

4.2.2. Product Information

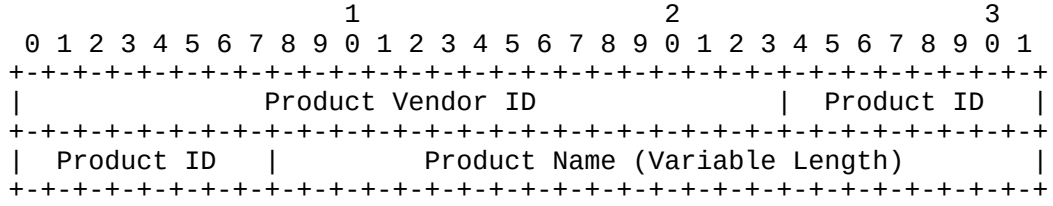
This PA-TNC Attribute Type contains identifying information about a product that implements the component specified in the PA Subtype field, as described in section 3.5. For example, if the PA Subtype is Anti-Virus, this attribute would contain information identifying an anti-virus product installed on the endpoint.

All Posture Collectors that implement any of the IETF Standard PA Subtypes defined in this specification MUST support sending this attribute type, at least for those PA subtypes. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. All Posture Validators that implement any of the IETF Standard PA Subtypes defined in this specification MUST support receiving this attribute type, at least for those PA subtypes. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 2. The value in the PA-TNC Attribute Length field will vary, depending on the length of the Product Name field. However, the value in the PA-TNC Attribute Length field MUST be at least 17 because this is the length of the fixed-length fields in the PA-TNC Attribute Header and the fixed-length fields in this attribute type. If the PA-TNC Attribute Length field is less than the size of these fixed-length fields, implementations SHOULD respond with an Invalid Parameter PA-TNC error code.

This attribute type includes both numeric and textual identifiers for the organization that created the product (the "product creator") and for the product itself. For automated processing, numeric identifiers are superior because they are less ambiguous and more efficient. However, numeric identifiers are only available if the product creator has assigned them. Therefore, a textual identifier is also included. This textual identifier has the additional benefit that it may be easier for humans to read (although this benefit is minimal since the primary purpose of this attribute is automated assessment).

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Product Vendor ID

This field contains the SMI Private Enterprise Number for the product creator. If the SMI PEN for the product creator is unknown or if the product creator does not have an SMI PEN, the Product Vendor ID field MUST be set to 0 and the identity of the product creator SHOULD be included in the Product Name along with the name of the product.

Product ID

This field identifies the product using a numeric identifier assigned by the product creator. If this Product ID value is unknown or if the product creator has not assigned such a value, this field MUST be set to 0. If the Product Vendor ID is 0, this field MUST be set to 0. In any case, the name of the product SHOULD be included in the Product Name field.

Note that a particular Product ID value (e.g., 635) will have completely different meanings depending on the Product Vendor ID. Each Product Vendor ID defines a different space of Product ID values. Product creators are encouraged to publish lists of Product ID values for their products.

Product Name

This variable-length field contains a UTF-8 [2] string identifying the product (e.g., "Symantec Norton AntiVirus(TM) 2008") in enough detail to unambiguously distinguish it from other products from the product creator. Products whose creator is known, but does not have a registered SMI Private Enterprise Number, SHOULD be represented using a combination of the creator name and full product name (e.g., "Ubuntu(R) IPtables" for the IPtables firewall in the Ubuntu distribution of Linux). If the product creator's SMI Private Enterprise Number is included in the Product Vendor ID field, the product creator's name may be omitted from this field.

The length of this field can be determined by starting with the value in the PA-TNC Attribute Length field in the PA-TNC Attribute Header and subtracting the size of the fixed-length fields in that header (12) and the size of the fixed-length fields in this attribute (5). If the PA-TNC Attribute Length field is less than the size of these fixed-length fields, implementations SHOULD respond with an Invalid Parameter PA-TNC error code.

4.2.3. Numeric Version

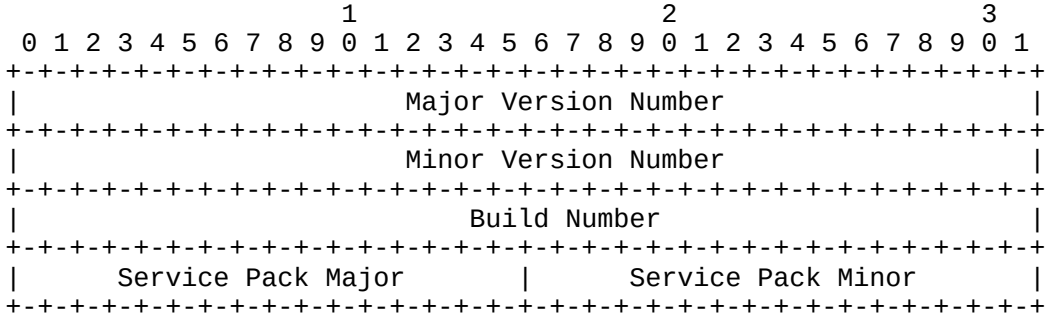
This PA-TNC Attribute Type contains numeric version information for a product on the endpoint that implements the component specified in the PA Subtype field, as described in section 3.5. For example, if the PA Subtype is Operating System, this attribute would contain numeric version information for the operating system installed on the endpoint. The version information in this attribute is associated with a particular product, so Posture Validators are expected to also possess the corresponding Product Information attribute when interpreting this attribute.

All Posture Collectors that implement the IETF Standard PA Subtype for Operating System SHOULD support sending this attribute type, at least for the Operating System PA subtype. Other Posture Collectors MAY support sending this attribute type. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. All Posture Validators that implement the IETF Standard PA Subtype for Operating System SHOULD support receiving this attribute type, at least for the Operating System PA subtype. Other Posture Validators MAY support receiving this attribute type. A Posture Validator that does not support receiving this attribute type SHOULD simply ignore attributes with this type. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 3. The value in the PA-TNC Attribute Length field MUST be 28. If the PA-TNC Attribute Length field is less than the size of these fixed-length fields, implementations SHOULD respond with an Invalid Parameter PA-TNC error code.

This attribute type includes numeric values for the product version information, enabling Posture Validators to do comparative operations on the version. Some Posture Collectors may not be able to determine some or all of this information for a product. However, this attribute can be especially useful for describing the version of the operating system, where numeric version numbers are generally available.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Major Version Number

This field contains the major version number for the product, if applicable. If unused or unknown, this field SHOULD be set to 0.

Minor Version Number

This field contains the minor version number for the product, if applicable. If unused or unknown, this field SHOULD be set to 0.

Build Number

This field contains the build number for the product, if applicable. This may provide more granularity than the minor version number, as many builds may occur leading up to an official release, and all these builds may share a single major and minor version number. If unused or unknown, this field SHOULD be set to 0.

Service Pack Major

This field contains the major version number of the service pack for the product, if applicable. If unused or unknown, this field SHOULD be set to 0.

Service Pack Minor

This field contains the minor version number of the service pack for the product, if applicable. If unused or unknown, this field SHOULD be set to 0.

4.2.4. String Version

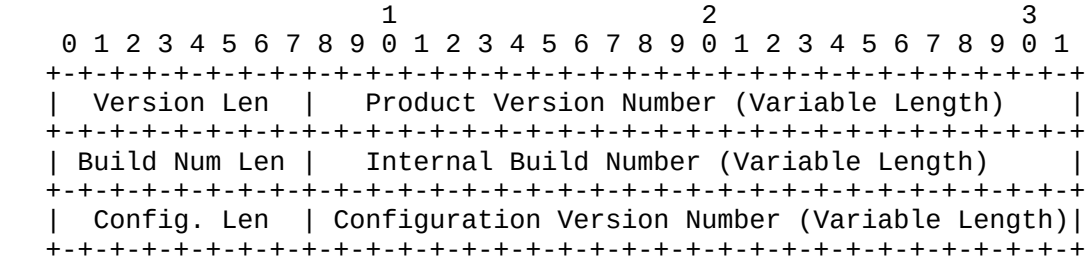
This PA-TNC Attribute Type contains string version information for a product on the endpoint that implements the component specified in the PA Subtype field, as described in section 3.5. For example, if the PA Subtype is Firewall, this attribute would contain string version information for a host-based firewall product installed on the endpoint (if any). The version information in this attribute is associated with a particular product, so Posture Validators are expected to also possess the corresponding Product Information attribute when interpreting this attribute.

All Posture Collectors that implement any of the IETF Standard PA Subtypes defined in this document MUST support sending this attribute type, at least for those PA subtypes. Other Posture Collectors MAY support sending this attribute type. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. All Posture Validators that implement any of the IETF Standard PA Subtypes defined in this document MUST support receiving this attribute type, at least for those PA subtypes. Other Posture Validators MAY support receiving this attribute type. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 4. The value in the PA-TNC Attribute Length field will vary, depending

on the length of the Component Version Number, Internal Build Number, and Configuration Version Number fields. However, the value in the PA-TNC Attribute Length field MUST be at least 15 because this is the length of the fixed-length fields in the PA-TNC Attribute Header and the fixed-length fields in this attribute type. If the PA-TNC Attribute Length field is less than the size of these fixed-length fields or does not match the length indicated by the sum of the fixed-length and variable-length fields, implementations SHOULD respond with an Invalid Parameter PA-TNC error code.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Version Len

This field defines the number of octets in the Product Version Number field. If the product version number is unavailable or unknown, this field MUST be set to 0 and the Product Version Number field will be zero length (effectively not present).

Product Version Number

This field contains a UTF-8 string identifying the version of the component (e.g., "1.12.23.114"). This field MUST be sized to fit the version string and MUST NOT include extra octets for padding or NUL character termination.

Various products use a wide range of different formats and semantics for version strings. Some use alphabetic characters, white space, and punctuation. Some consider version "1.21" to be later than version "1.3" and some earlier. Therefore, the syntax and semantics of this string are not defined.

Build Num Len

This field defines the number of octets in the Internal Build Number field. For products where the internal build number is unavailable or unknown, this field MUST be set to 0 and the Internal Build Number field will be zero length (effectively not present).

Internal Build Number

This field contains a UTF-8 string identifying the engineering build number of the product. This field MUST be sized to fit the build number string and MUST NOT include extra octets for padding or NUL character termination. The syntax and semantics of this string are not defined.

Config. Len

This field defines the number of octets in the Configuration Version Number field. If the configuration version number is unavailable or unknown, this field MUST be set to 0 and the Configuration Version Number field will be zero length (effectively not present).

Configuration Version Number

This field contains a UTF-8 string identifying the version of the configuration used by the component. This version SHOULD represent the overall configuration version even if several configuration policy files or settings are used. Posture Collectors MAY include multiple version numbers in this single string if a single version is not practical. This field MUST be sized to fit the version string and MUST NOT include extra octets for padding or NUL character termination.

Various products use a wide range of different formats for version strings. Some use alphabetic characters, white space, and punctuation. Some consider version "1.21" to be later than version "1.3" and some earlier. In addition, some Posture Collectors may place multiple configuration version numbers in this single string. Therefore, the syntax and semantics of this string are not defined.

4.2.5. Operational Status

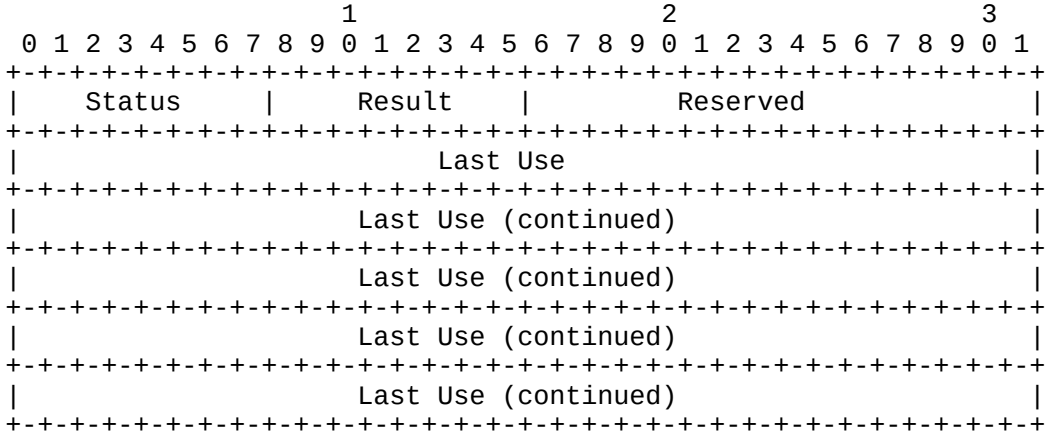
This PA-TNC Attribute Type describes the operational status of a product that can implement the component specified in the PA Subtype field, as described in section 3.5. For example, if the PA Subtype is

Anti-Spyware, this attribute would contain information about the operational status of a host-based anti-spyware product that may or may not be installed on the endpoint.

Posture Collectors that implement the IETF Standard PA Subtype for Operating System or VPN MAY support sending this attribute type for those PA subtypes. Posture Collectors that implement other IETF Standard PA Subtypes defined in this specification SHOULD support sending this attribute type for those PA subtypes. Other Posture Collectors MAY support sending this attribute type. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. Posture Validators that implement the IETF Standard PA Subtype for Operating System or VPN MAY support receiving this attribute type, at least for those PA subtypes. Posture Validators that implement other IETF Standard PA Subtypes defined in this specification SHOULD support receiving this attribute type, at least for those PA subtypes. Other Posture Validators MAY support receiving this attribute type. A Posture Validator that does not support receiving this attribute type SHOULD simply ignore attributes with this type. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 5. The value in the PA-TNC Attribute Length field MUST be 36. If the PA-TNC Attribute Length field does not have this value, implementations SHOULD respond with an Invalid Parameter PA-TNC error code.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Status

This field gives the operational status of the product. The following table lists the values currently defined for this field.

Value	Description
0	Unknown or other
1	Not installed
2	Installed but not operational
3	Operational

If a Posture Validator receives a value for this field that it does not recognize, it SHOULD treat this value as equivalent to the value 0.

Result

This field contains the result of the last use of the product. The following table lists the values currently defined for this field.

Value	Description
0	Unknown or other
1	Successful use with no errors detected
2	Successful use with one or more errors detected
3	Unsuccessful use (e.g., aborted)

Posture Collectors SHOULD set this field to 0 if the Status field contains a value of 1 (Not installed) or 2 (Installed but not operational). If a Posture Validator receives a value for this field that it does not recognize, it SHOULD treat this value as equivalent to the value 0.

Reserved

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

Last Use

This field contains the date and time of the last use of the component. The Last Use date and time MUST be represented as an RFC 3339 [4] compliant ASCII string in Coordinated Universal Time (UTC) time with the additional restrictions that the 't' delimiter and the 'z' suffix MUST be capitalized and fractional seconds (time-secfrac) MUST NOT be included.

This field conforms to the date-time ABNF production from section 5.6 of RFC 3339 with the above restrictions. Leap seconds are permitted and Posture Validators MUST support them.

The last use string MUST NOT be NUL terminated or padded in any way. If the last use time is not known, not applicable, or cannot be represented in this format, the Posture Collector MUST set this field to the value "0000-00-00T00:00:00Z" (allowing this field to be fixed length). Note that this particular reserved value is NOT a valid RFC 3339 date and time and MUST NOT be used for any other purpose in this field.

This encoding produces a string that is easy to read, parse, and interpret. The format (more precisely defined in RFC 3339) is YYYY-MM-DDTHH:MM:SSZ, resulting in one and only one representation for each second in UTC time from year 0000 to year 9999. For example, 9:05:00AM EST (GMT-0500) on January 19, 1995 can be represented as "1995-01-19T14:05:00Z". The length of this field is always 20 octets.

4.2.6. Port Filter

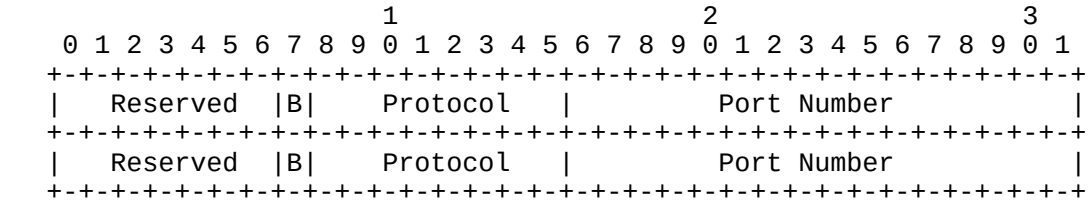
This PA-TNC Attribute Type provides the list of port numbers and associated protocols (e.g., TCP and UDP) that are currently blocked or allowed by a host-based firewall on the endpoint.

Posture Collectors that implement the IETF Standard PA Subtype for Firewall or VPN SHOULD support sending this attribute type for those PA subtypes. Posture Collectors that implement other IETF Standard PA Subtypes defined in this specification MUST NOT support sending this attribute type for those PA subtypes. Other Posture Collectors MAY support sending this attribute type, if it is appropriate to their PA subtype. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. Posture Validators that implement the IETF Standard PA Subtype for Firewall or VPN SHOULD support receiving this attribute type, at least for those PA subtypes. Posture Validators that implement other IETF Standard PA Subtypes defined in this specification MUST NOT support receiving this attribute type for those PA subtypes. Other Posture Validators MAY support receiving this attribute type. A Posture Validator that does not support receiving this attribute type SHOULD simply ignore attributes with this type. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 6.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.

Note that this diagram shows two Protocol/Port Number pairs. The actual number of Protocol/Port Number pairs included in a Port Filter attribute can vary from one to a large number (limited only by the maximum message and length supported by the underlying PT protocol). However, each Port Filter attribute MUST contain at least one Protocol/Port Number pair. Because the length of a Protocol/Port Number pair with the Reserved field and B flag is always 4 octets, the number of Protocol/Port Number pairs can be easily computed using the PA-TNC Attribute Length field by subtracting the number of octets in the PA-TNC Attribute Header and dividing by 4. If the PA-TNC Attribute Length field is invalid, Posture Validators SHOULD respond with an Invalid Parameter PA-TNC error code.



Reserved

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

B Flag (Blocked or Allowed Port)

This single-bit field indicates whether the following port is blocked or allowed. This bit MUST be set to 1 if the protocol and port combination is blocked. Otherwise, this field MUST be set to 0. This field was provided to allow for more abbreviated reporting of the port filtering policy (e.g., when all ports are blocked except a few, the Posture Collector can just list the few that are allowed).

Posture Collectors MUST NOT provide a mixed list of blocked and non-blocked ports for a particular protocol. To be more precise, a Posture Collector MUST NOT include two Protocol/Port Number pairs in a single Port Filter attribute where the protocol number is the same but the B flag is different. Also, Posture Collectors MUST NOT list the same Protocol and Port Number combination twice in a Port List attribute.

Posture Collectors MAY list all blocked ports for one protocol and all allowed ports for a different protocol in a single Port List attribute, using the B flag to indicate whether each entry is blocked. For example, a Posture Collector might list all the blocked TCP ports but only list the allowed UDP ports. However, it MUST NOT list some blocked TCP ports and some other allowed TCP ports.

Protocol

This field contains the transport protocol number (e.g., tcp is 6) being blocked or allowed. The values used in this field are the same ones used in the IPv4 Protocol and IPv6 Next Header fields. The IANA already maintains the Assigned Internet Protocol Numbers registry of these values for use in this field.

Port Number

This field contains the transport protocol (e.g., tcp) port number being blocked or allowed. The values used in this field are specific to the protocol identified by the Protocol field. The IANA maintains registries for well-known and user-requested TCP and UDP port numbers for use in this field.

4.2.7. Installed Packages

This PA-TNC Attribute Type contains a list of the installed packages that comprise a product on the endpoint that implements the component specified in the PA Subtype field, as described in section 3.5. This allows a Posture Validator to check which packages are installed for a particular product and which versions of those packages are installed.

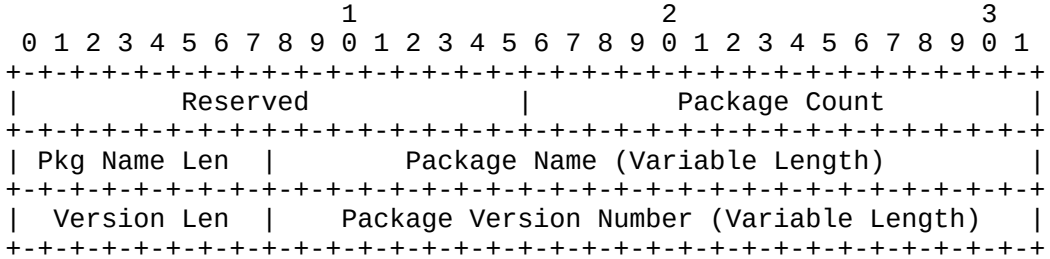
Posture Collectors that implement any of the IETF Standard PA Subtypes defined in this document SHOULD support sending this attribute type for those PA subtypes. Other Posture Collectors MAY support sending this attribute type, if it is appropriate to their PA subtype. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. Posture Validators that implement any of the IETF Standard PA Subtypes defined in this document SHOULD support receiving this attribute type, at least for those PA subtypes. Other Posture Validators MAY support receiving this attribute type. A Posture Validator that does not support receiving this attribute type SHOULD simply ignore attributes with this type. Posture Validators MUST NOT send this attribute type.

This attribute type can be quite long, especially for the Operating System PA subtype. This can cause problems, especially with 802.1X and other limited transport protocols. Therefore, Posture Collectors SHOULD NOT send this attribute unless specifically requested to do so using the Attribute Request attribute or otherwise configured to do so. Also, Posture Validators SHOULD NOT request this attribute unless the transport protocol in use can support the large amount of data that may be sent in response.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 7. The value in the PA-TNC Attribute Length field will vary, depending on the number of packages and the length of the Package Name and Package Version Number fields for those packages. However, the value in the PA-TNC Attribute Length field MUST be at least 16 because this is the length of the fixed-length fields in the PA-TNC Attribute Header and the fixed-length fields in this attribute type. If the PA-TNC Attribute Length field is less than the size of these fixed-length fields or does not match the length indicated by the sum of the fixed-length and variable-length fields, implementations SHOULD respond with an Invalid Parameter PA-TNC error code.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.

Note that this diagram shows an attribute containing information on one package. The actual number of package descriptions included in an Installed Packages attribute is indicated by the Package Count field. This value may vary from zero to a large number (up to 65535, if the underlying PT protocol can support that many). If this number is not sufficient, specialized patch management software should be employed that can simply report compliance with a pre-established patch policy.



Reserved

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

Package Count

This field is an unsigned 16-bit integer that indicates the number of packages listed in this attribute. For each package so indicated, a Pkg Name Len, Package Name, Version Len, and Package Version Number field is included in the attribute.

Pkg Name Len

This field is an unsigned 8-bit integer that indicates the length of the Package Name field in octets. This field may be zero if a Package Name is not available.

Package Name

This field contains the name of the package associated with the product. This field is a UTF-8 encoded character string whose octet length is given by the Pkg Name Len field. This field MUST NOT include extra octets for padding or NUL character termination. The syntax and semantics of this name are not specified in this document, since they may vary across products and/or operating systems. Posture Collectors MAY list two packages with the same name in a single Installed Packages attribute. The meaning of doing so is not defined here.

Version Len

This field is an unsigned 8-bit integer that indicates the length of the Package Version Number field in octets. This field may be zero if a Package Version Number is not available.

Package Version Number

This field contains the version string for the package named in the previous Package Name field. This field is a UTF-8 encoded character string whose octet length is given by the Version Len field. This field MUST NOT include extra octets for padding or NUL character termination. The syntax and semantics of this version string are not specified in this document, since they may vary across products and/or operating systems. Posture Collectors

MAY list two packages with the same Package Version Number (and even the same Package Name and Package Version Number) in a single Installed Packages attribute. The meaning of doing so is not defined here.

4.2.8. PA-TNC Error

This PA-TNC Attribute Type contains an error code and supplemental information regarding an error pertaining to PA-TNC.

All Posture Collectors and Posture Validators that implement any of the IETF Standard PA Subtypes defined in this specification MUST support sending and receiving this attribute type, at least for those PA subtypes.

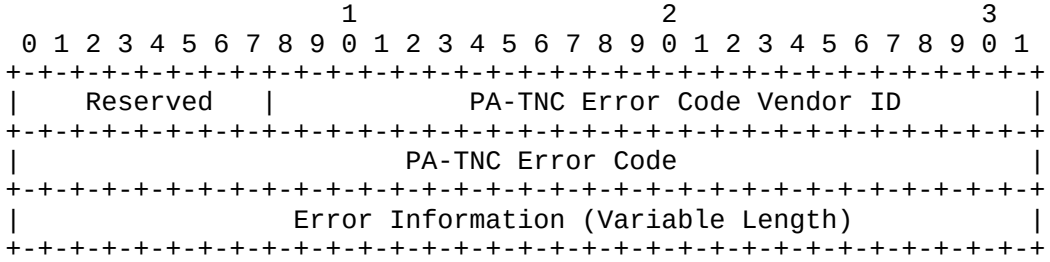
For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 8. The value in the PA-TNC Attribute Length field will vary, depending on the length of the Error Information field. However, the value in the PA-TNC Attribute Length field MUST be at least 20 because this is the length of the fixed-length fields in the PA-TNC Attribute Header and the fixed-length fields in this attribute type.

A PA-TNC error code SHOULD be sent with the same PA Message Vendor ID and PA Subtype used by the PA-TNC message that caused the error so that the error code is sent to the party who sent the offending PA-TNC message. Other measures (such as setting PB-TNC's EXCL flag and Posture Collector Identifier or Posture Validator Identifier fields) SHOULD also be taken to attempt to ensure that only the party who sent the offending message receives the error.

When a PA-TNC error code is received, the recipient MUST NOT respond with a PA-TNC error code because this could result in an infinite loop of errors. Instead, the recipient MAY log the error, modify its behavior to attempt to avoid the error (attempting to avoid loops or long strings of errors), ignore the error, terminate the assessment, or take other action as appropriate (as long as it is consistent with the requirements of this specification).

Posture Validators MUST NOT include this attribute type in an Attribute Request attribute. It does not make sense for a Posture Validator to request that a Posture Collector send a PA-TNC Error attribute.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Reserved

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

PA-TNC Error Code Vendor ID

This field contains the SMI Private Enterprise Number for the organization that defined the PA-TNC Error Code that is being used in the attribute. For IETF Standard PA-TNC Error Code values this field MUST be set to zero (0).

PA-TNC Error Code

This field contains the PA-TNC Error Code being reported in this attribute. Note that a particular PA-TNC Error Code value will have completely different meanings depending on the PA-TNC Error Code Vendor ID. Each PA-TNC Error Code Vendor ID defines a different space of PA-TNC Error Code values. Posture Collectors and Posture Validators MUST NOT require support for particular vendor-specific PA-TNC Error Codes and MUST interoperate with other parties despite any differences in the set of vendor-specific PA-TNC Error Codes supported (although they MAY permit administrators to configure them to require support for specific PA-TNC Error Codes).

When the PA-TNC Error Code Vendor ID is set to zero (0), the PA-TNC Error Code is an IETF Standard PA-TNC Error Code. IANA maintains a registry of PA-TNC Error Codes. Entries in this registry are added by Expert Review with Specification Required, following the guidelines in section 7.

The following table lists the IETF Standard PA-TNC Error Codes defined in this specification:

Integer	Description
0	Reserved
1	Invalid Parameter
2	Version Not Supported
3	Attribute Type Not Supported

The next few subsections of this document provide detailed definitions of these error codes.

Error Information

This field provides additional context for the error. The contents of this field vary based on the PA-TNC Error Code Vendor ID and PA-TNC Error Code. Therefore, whenever a PA-TNC Error Code is defined, the format of this field for that error code must also be defined. The definitions of IETF Standard PA-TNC Error Codes on the next few pages provide good examples of such definitions.

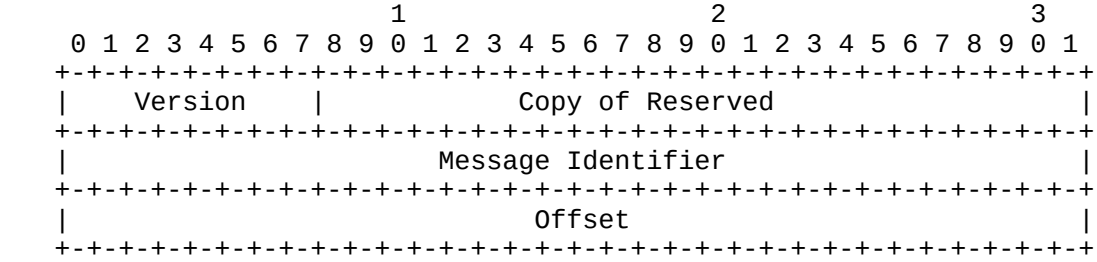
The length of this field can be determined by the recipient using the PA-TNC Attribute Length field by subtracting the length of the fixed-length fields in the PA-TNC Attribute Header and the fixed-length fields in this attribute.

4.2.8.1. Invalid Parameter Error Code

The Invalid Parameter error code is an IETF Standard PA-TNC Error Code (value 1) that indicates that the sender of this error code has detected an invalid value in a PA-TNC message sent by the recipient of this error code in the current assessment.

For this error code, the Error Information field contains the first 8 octets of the PA-TNC message that contained the invalid parameter and an offset indicating the position within that message of the invalid parameter.

The following diagram illustrates the format and contents of the Error Information field for this error code. The text after this diagram describes the fields shown here.



Version

This field MUST contain an exact copy of the Version field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Copy of Reserved

This field MUST contain an exact copy of the Reserved field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Message Identifier

This field MUST contain an exact copy of the Message Identifier field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Offset

This field MUST contain an octet offset from the start of the PA-TNC Message Header of the PA-TNC message that caused this error to the start of the value that caused this error. For instance, if the first PA-TNC attribute in the message had an invalid PA-TNC Attribute Length (e.g., 0), this value would be 16.

4.2.8.2. Version Not Supported Error Code

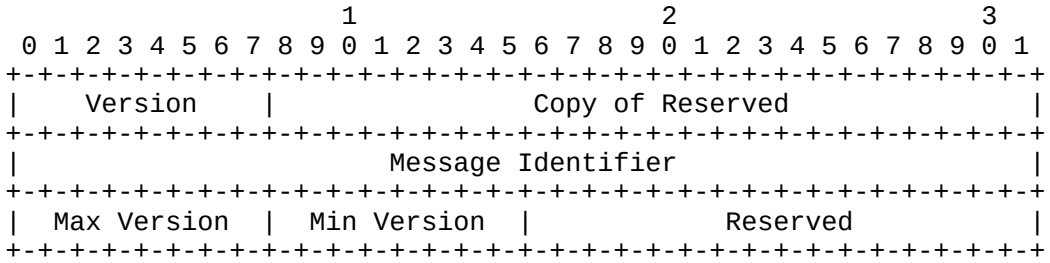
The Version Not Supported error code is an IETF Standard PA-TNC Error Code (value 2) that indicates that the sender of this error code does not support the PA-TNC version number included in the PA-TNC Message Header of a PA-TNC message sent by the recipient of this error code in the current assessment.

For this error code, the Error Information field contains the first 8 octets of the PA-TNC message that contained the unsupported version as well as Max Version and Min Version fields that indicate which PA-TNC version numbers are supported by the sender of the error code.

The sender MUST support all PA-TNC versions between the Min Version and the Max Version, inclusive (i.e., including the Min Version and the Max Version). When possible, recipients of this error code SHOULD send future messages to the Posture Collector or Posture Validator that originated this error message with a PA-TNC version number within the stated range.

Any party that is sending the Version Not Supported error code MUST include that error code as the only PA-TNC attribute in a PA-TNC message with version number 1. All parties that send PA-TNC messages MUST be able to properly process a message that meets this description, even if they cannot process any other aspect of PA-TNC version 1. This ensures that a PA-TNC version exchange can proceed properly, no matter what versions of PA-TNC the parties implement.

The following diagram illustrates the format and contents of the Error Information field for this error code. The text after this diagram describes the fields shown here.



Version

This field MUST contain an exact copy of the Version field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Copy of Reserved

This field MUST contain an exact copy of the Reserved field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Message Identifier

This field MUST contain an exact copy of the Message Identifier field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Max Version

This field MUST contain the maximum PA-TNC version supported by the sender of this error code.

Min Version

This field MUST contain the minimum PA-TNC version supported by the sender of this error code.

Reserved

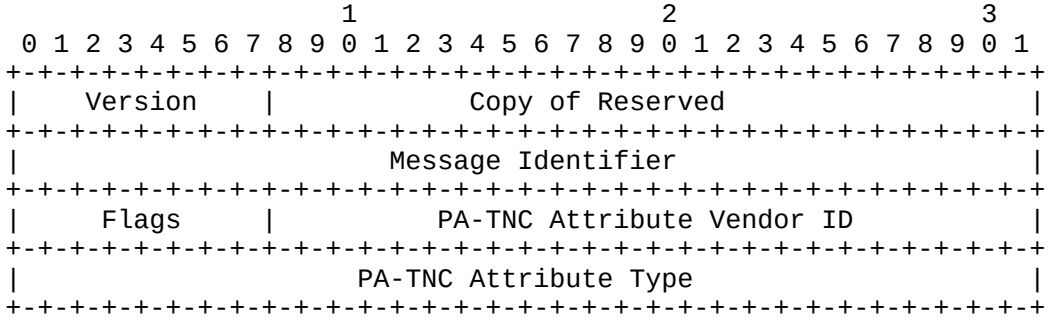
Reserved for future use. This field MUST be set to 0 on transmission and ignored upon reception.

4.2.8.3. Attribute Type Not Supported Error Code

The Attribute Type Not Supported error code is an IETF Standard PA-TNC Error Code (value 3) that indicates that the sender of this error code does not support the PA-TNC Attribute Type included in the Error Information field. This PA-TNC Attribute Type was included in a PA-TNC message sent by the recipient of this error code in the current assessment.

For this error code, the Error Information field contains the first 8 octets of the PA-TNC message that contained the unsupported attribute type as well as a copy of the attribute type that caused the problem.

The following diagram illustrates the format and contents of the Error Information field for this error code. The text after this diagram describes the fields shown here.



Version

This field MUST contain an exact copy of the Version field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Copy of Reserved

This field MUST contain an exact copy of the Reserved field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Message Identifier

This field MUST contain an exact copy of the Message Identifier field in the PA-TNC Message Header of the PA-TNC message that caused this error.

Flags

This field MUST contain an exact copy of the Flags field in the PA-TNC Attribute Header of the PA-TNC attribute that caused this error.

PA-TNC Attribute Vendor ID

This field MUST contain an exact copy of the PA-TNC Attribute Vendor ID field in the PA-TNC Attribute Header of the PA-TNC attribute that caused this error.

PA-TNC Attribute Type

This field MUST contain an exact copy of the PA-TNC Attribute Type field in the PA-TNC Attribute Header of the PA-TNC attribute that caused this error.

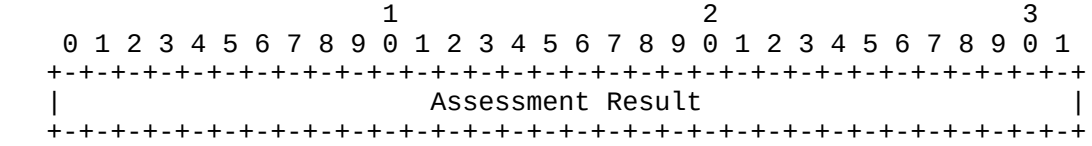
4.2.9. Assessment Result

This PA-TNC attribute contains the final assessment result from a particular Posture Validator. This attribute might be returned to a Posture Collector for information purposes such as when an endpoint is compliant. Similarly, the Assessment Result attribute could be sent to indicate a non-compliant result where specific actions are needed to bring an endpoint into compliance with the network's policies. These actions could be defined in other PA-TNC attributes such as Remediation Instructions sent to the Posture Collector.

All Posture Collectors that support an IETF Standard PA Subtype defined in this specification SHOULD support receiving and processing the Assessment Result attribute. All Posture Validators that implement an IETF Standard PA Subtype defined in this specification SHOULD support sending the Assessment Result attribute.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 9.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Assessment Result

This 32-bit field MUST contain one of the following values;

Value	Description
0	Posture Validator assessed the endpoint component to be compliant with policy.
1	Posture Validator assessed the endpoint component to be non-compliant with policy but the difference from compliant was minor.

2 Posture Validator assessed the endpoint component to be non-compliant with policy and the assessed difference was very significant.

3 Posture Validator was unable to determine policy compliance of an endpoint component due to an error.

4 Posture Validator was unable to determine whether the assessed endpoint component was compliant with policy based on the attributes provided by the Posture Collector.

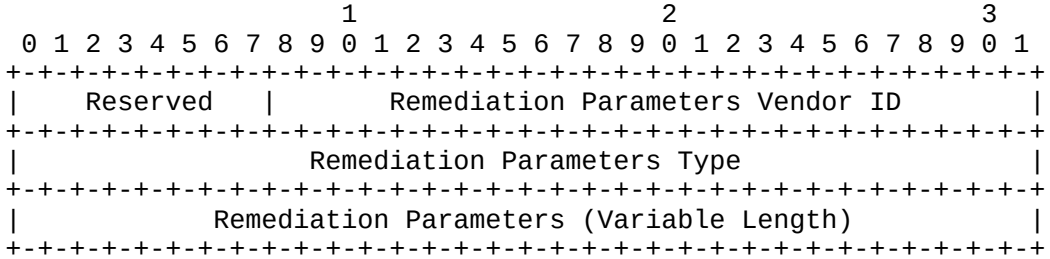
4.2.10. Remediation Instructions

This PA-TNC attribute sent by the Posture Validator to the Posture Collector contains remediation instructions for updating a particular component to make the endpoint compliant with the assessment policies. A Posture Validator might choose to send more than one Remediation Instructions attribute in some circumstances (e.g., both a URI and a human-readable message are necessary) to remediate one or more components. This attribute supports the inclusion of either an IETF standard or vendor-specific remediation instruction.

All Posture Collectors that implement an IETF Standard PA Subtype defined in this specification SHOULD support receiving and processing the Remediation Instructions attribute. All Posture Validators that implement an IETF Standard PA Subtype defined in this specification SHOULD support sending this attribute type. Posture Collectors and Posture Validators supporting other non-IETF standard components MAY support this attribute.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 10.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Reserved (8 bits)

The Reserved bits MUST be set to 0 on transmission and ignored on reception.

Remediation Parameters Vendor ID (24 bits)

The Remediation Parameters Vendor ID field identifies a vendor by using the SMI Private Enterprise Number (PEN). Any organization can receive its own unique PEN from IANA, the Internet Assigned Numbers Authority. The Remediation Parameters Vendor ID qualifies the Remediation Parameters Type field so that each vendor has 2^32 separate Remediation Parameters Types available for its use. Remediation Parameters Types standardized by the IETF are always used with the value zero (0) in this field.

Remediation Parameters Type (32 bits)

The Remediation Parameters Type field identifies the different types of remediation instructions that can be contained in the Remediation Parameters field. IANA maintains a registry of PA-TNC Remediation Parameters Types. Entries in this registry are added by Expert Review with Specification Required, following the guidelines in section 7. A list of IETF Standard PA-TNC Remediation Parameters Types defined in this specification appears later in this section.

New vendor-specific remediation instructions can be created by adding new Remediation Parameters Types (those used with a non-zero Remediation Parameters vendor ID) without IETF or IANA involvement. Posture Collectors and Posture Validators MUST NOT require support for particular vendor-specific PA-TNC Remediation Parameters Types and MUST interoperate with other parties despite any differences in the set of vendor-specific PA-TNC Remediation Parameters Types supported (although they MAY permit administrators to configure them to require support for specific PA-TNC remediation parameter types).

The following table lists the IETF Standard PA-TNC Remediation Parameters Type values defined in this specification:

Integer	Description
0	Reserved
1	Remediation URI
2	Remediation String

The next few subsections of this document provide detailed definitions of the contents of the Remediation Parameters field used with each Remediation Parameter Type.

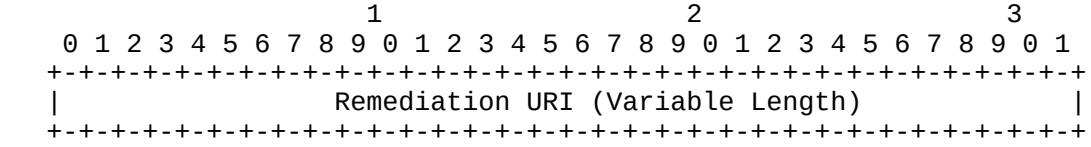
Remediation Parameters (variable length)

The Remediation Parameters field contains the actual remediation instructions for the Posture Collector.

4.2.10.1. Remediation URI Parameters Type

The Remediation URI Parameters Type is an IETF Standard Remediation Parameters Type (value 1) that indicates that the sending Posture Validator is providing a URI to instructions on how to remediate the endpoint.

The following diagram illustrates the format and contents of the Remediation Parameters field when carrying a Remediation URI parameter. The text after this diagram describes the fields shown here.



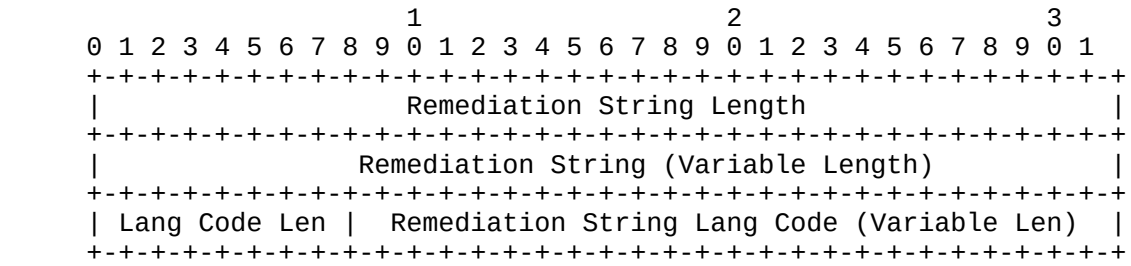
Remediation URI

The Remediation URI field MUST contain a URI, as described in RFC 3986 [7]. This URI SHOULD contain instructions to update a particular component so that it might result in the component being compliant with the policies in future assessments. Posture Collectors should validate that the URI and instructions come from a trustworthy source to avoid being tricked into performing damaging actions (see security considerations).

4.2.10.2. Remediation String Parameters Type

The Remediation String Parameters Type is an IETF Standard Remediation Parameters Type (value 2) that indicates that the sending Posture Validator is providing a human-readable string containing instructions on how to remediate the endpoint.

The following diagram illustrates the format and contents of the Remediation Parameters field when the carrying a Remediation String parameter. The text after this diagram describes the fields shown here.



Remediation String Length

The Remediation String Length contains the length of the Remediation String field in octets.

Remediation String

The Remediation String field MUST contain a UTF-8 encoded string. This string contains human-readable instructions for remediation that MAY be displayed to the user by the Posture Collector. NUL termination MUST NOT be included. If a Posture Collector receives a Remediation String that does contain a NUL termination, it SHOULD send an Invalid Parameter error code.

Lang Code Len (Remediation String Language Code Length)

The Lang Code Len field contains the length of the Remediation String Language Code field in octets.

Remediation String Lang Code

The Remediation String Lang(uage) Code field contains a US-ASCII string composed of a well-formed RFC 4646 [6] language tag that indicates the language(s) used in the Remediation String in the Remediation Parameters field. A zero-length string MAY be sent for this field (essentially omitting this field) to indicate that the language code for the remediation string is not known.

4.2.11. Forwarding Enabled

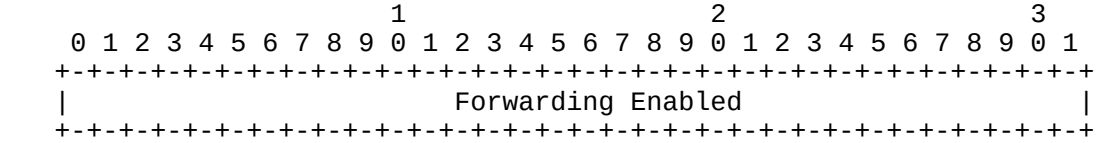
This PA-TNC attribute indicates whether the endpoint is forwarding traffic between interfaces. Endpoints that forward traffic between networks connected to multiple network interfaces may be considered non-compliant (and a security risk) in some enterprise network deployments. For example, an endpoint with multiple connected network interfaces might allow traffic from an interface connected to a public network to be forwarded through another interface carrying a VPN session to a protected enterprise network. This attribute is

currently envisioned to be specific to reporting posture for the operating system component; however, could be useful for other future types of components.

Posture Collectors that implement the IETF Standard PA Subtype for Operating System SHOULD support sending the Forwarding Enabled attribute. Posture Collectors that do not implement the Operating System PA Subtype defined in this specification SHOULD NOT send the Forwarding Enabled attribute unless it is appropriate to their PA Subtype. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. Posture Validators that implement the IETF Standard PA Subtype for Operating System SHOULD support receiving the Forwarding Enabled attribute type. Posture Validators supporting components other than Operating System MAY support receiving this attribute type if it is appropriate to their PA Subtype. A Posture Validator that does not support receiving this attribute type SHOULD simply ignore attributes with this type. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 11.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Forwarding Enabled

This 32-bit field MUST contain one of the following values;

Value	Description
0	Disabled - Endpoint is not forwarding traffic.
1	Enabled - Endpoint is forwarding traffic.
2	Unknown - Unable to determine whether endpoint is forwarding traffic

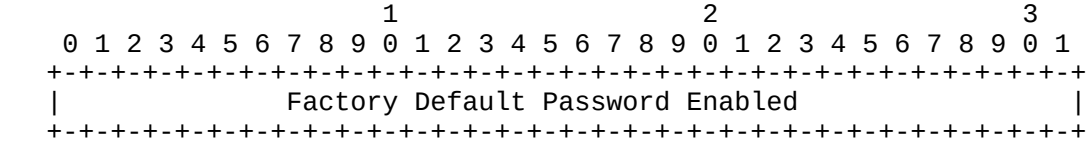
4.2.12. Factory Default Password Enabled

This PA-TNC attribute indicates whether the endpoint has a factory default password enabled for use. Some types of endpoints include a default static password for used to gain privileged access to the endpoint. If this password is not changed or disabled before the endpoint is accessible on the network, it's often easy to compromise the endpoint.

Posture Collectors that implement the IETF Standard PA Subtype for Operating System SHOULD support sending the Factory Default Password Enabled attribute. Posture Collectors that implement other IETF Standard PA Subtypes defined in this specification SHOULD NOT support sending this attribute type for those PA subtypes. Other Posture Collectors MAY support sending this attribute type, if it is appropriate to their PA subtype. Whether a particular Posture Collector actually sends this attribute type SHOULD still be governed by local privacy and security policies. Posture Validators that implement the IETF Standard PA Subtype for Operating System SHOULD support receiving the Factory Default Password Enabled attribute. Other Posture Validators MAY support receiving this attribute type. A Posture Validator that does not support receiving this attribute type SHOULD simply ignore attributes with this type. Posture Validators MUST NOT send this attribute type.

For this attribute type, the PA-TNC Attribute Vendor ID field MUST be set to zero (0) and the PA-TNC Attribute Type field MUST be set to 12.

The following diagram illustrates the format and contents of the Attribute Value field for this attribute type. The text after this diagram describes the fields shown here.



Factory Default Password Enabled

This 32-bit field MUST contain one of the following values;

Value	Description
0	Endpoint does not have factory default password enabled.
1	Endpoint has a factory default password enabled.

4.3. Vendor-Defined Attributes

This section discusses the use of vendor-defined attributes within PA-TNC. The PA-TNC protocol was designed to allow for vendor-defined attributes to be used as a replacement where a standard attribute could be used. In some cases, even the standard attributes allow for vendor-defined information to be included. It is envisioned that over time as particular vendor-defined attributes become popular, an equivalent standard attribute could be added allowing for broader interoperability.

This specification does not define vendor-defined attributes, but rather highlights how such attributes can be used with PA-TNC without the potential for namespace collisions or misinterpretations. In order to avoid collisions, PA-TNC uses the well-established SMI Private Enterprise Numbers as vendor IDs to define separate namespaces for important fields within a PA-TNC message. For example, to ensure the uniqueness of attribute types while providing for vendor extensions, vendor-defined attribute types include the vendor's unique vendor ID, to indicate the intended namespace for the attribute type, followed by the attribute type. IETF Standard PA-TNC Attribute Types use a vendor ID of zero (0).

SMI Private Enterprise Numbers are used to provide a separate identifier space for each vendor. The IANA provides a registry for SMI Private Enterprise Numbers. Any organization (including non-profit organizations, governmental bodies, etc.) can obtain one of these numbers at no charge, and thousands of organizations have done so. Within this document, SMI Private Enterprise Numbers are known as "vendor IDs".

5. Security Considerations

This section discusses the major potential types of security threats relevant to the PA-TNC message protocol. It is envisioned that additional attribute types could be defined in the future to facilitate the exchange of security capabilities, keys, and security protected attributes if future use cases are adopted that require such protections.

5.1. Trust Relationships

In order to understand where security countermeasures are necessary, this section starts with a discussion of where the TNC architecture envisions some trust relationships between the processing elements of the PA-TNC protocol. The following subsections discuss the trust properties associated with each portion of the NEA reference model directly involved with the processing of the PA-TNC protocol.

5.1.1. Posture Collector

The Posture Collectors are trusted by Posture Validators to:

- Collect valid information about the component type associated with the Posture Collector
- Report upon collected information consistent with local security and privacy policies
- Accurately report information associated with the type of component for the PA-TNC message
- Not act maliciously to the Posture Broker Server and Posture Validators, including attacks such as denial of service

5.1.2. Posture Validator

The Posture Validators are trusted by Posture Collectors to:

- Only request information necessary to assess the security state of the endpoint
- Make assessment decisions based on deployer-defined policies
- Discard collected information consistent with data retention and privacy policies
- Not act maliciously to the Posture Broker Server and Posture Collectors, including attacks such as denial of service
- Not send malicious remediation instructions that do not fix or that cause damage to the endpoint

5.1.3. Posture Broker Client, Posture Broker Server

The Posture Broker Client and Posture Broker Server are trusted by the Posture Collector and Posture Validator to:

- Provide a reliable transport for PA-TNC messages
- Deliver messages for a particular PA Subtype only to those Posture Collectors and Posture Validators that have registered for them
- Not disclose any provided attributes to unauthorized parties

- Not act maliciously to drop messages, duplicate messages, or flood Posture Collectors and Posture Validators with unnecessary messages

- Not observe, fabricate, or alter the contents of a PA-TNC message

- Properly place Posture Collector and Posture Validator identifiers into the PB-TNC protocol, deliver those identifiers to Posture Collectors and Posture Validators as needed, and manage exclusive delivery to a particular Posture Collector or Posture Validator when requested

- Properly expose authentication information from PT security so that Posture Collectors and Posture Validators can use the peer's identity information to safely make policy decisions

5.2. Security Threats

Beyond the trusted relationships assumed in section 5.1, the PA-TNC protocol faces a number of potential security attacks that could require security countermeasures.

Generally, the PA-TNC protocol relies upon the underlying PT protocol's security to protect the messages from attack when traveling over the network. Once the message resides on the Posture Broker Client or Posture Broker Server, the posture brokers are trusted to properly and safely deliver the messages to the appropriate Posture Collectors and Posture Validators.

5.2.1. Attribute Theft

When PA-TNC messages are sent over unprotected network links or spanning local software stacks that are not trusted, the contents of the PA-TNC 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. Attributes 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 (e.g., certain endpoints are lacking patches, or particular sub-networks have more lenient policies).

PA-TNC attributes are not intended to carry privacy-sensitive information, but should some exist in a message, the adversary could come into possession of the information, which could be used for

financial gain. Therefore, it is important that PT provide strong confidentiality protection to protect the message from eavesdroppers when being sent between the Posture Transport Client and Posture Transport Server.

5.2.2. Message Fabrication

Attackers on the network or present within the NEA system could introduce fabricated PA-TNC messages intending to trick or create a denial of service against aspects of an assessment. For example, an adversary could attempt to send a falsified set of remediation instructions using the Remediation URI support in hopes of the Posture Collector automatically following the instructions. Posture Collectors need to ensure that any requests to take actions on the endpoint (such as remediation instructions) received from Posture Validators are authentic and trustworthy using strong authentication and integrity protections offered by PT. Posture Collectors should not blindly follow remediation instructions received from a trusted NEA Server. At least for patches and other potentially dangerous actions, Posture Collectors should validate these actions (e.g., via user confirmation) before proceeding.

Such an attack could occur if an active attacker launches a man-in-the-middle (MitM) attack by proxying the PA-TNC messages and was able to replace undesired messages with ones easing future attack upon the endpoint. Consider a scenario where PT security protection is not used and the Posture Broker Server proxies all assessment traffic to a remote Posture Broker Server. The proxy could eavesdrop and replace assessment results attributes, tricking the endpoint into thinking it has passed an assessment, when in fact it has not and requires remediation. Because the Posture Collector has no way to verify that attributes were actually created by an authentic Posture Validator, it is unable to detect the falsified attribute or message. Therefore, it is important that PT provides strong authentication and integrity protection.

5.2.3. Attribute Modification

This attack could allow an active attacker capable of intercepting a message to modify a PA-TNC message 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.

For example, an attacker might wish to change the contents of the firewall component’s version string attribute to disguise the fact that the firewall is running an old, vulnerable version. The

attacker would change the version string sent by the firewall Posture Collector to the current version number, so the Posture Validator's assessment passes while leaving the endpoint vulnerable to attack. Similarly, an attacker could achieve widespread denial of service by altering large numbers of assessments' version string attributes to an old value so they repeatedly fail assessments even after a successful remediation. Upon receiving the lower value, the Posture Validator would continue to believe that the endpoint is running old, potentially vulnerable versions of the firewall that does not meet network compliance policy, so therefore the endpoint would not be allowed to join the network. Use of a PT protocol providing strong integrity protection and authentication is essential as countermeasures to these attacks.

5.2.4. Attribute Replay

Another potential attack against an unprotected PA-TNC message attribute exchange is to exploit the lack of a strong binding between the attributes sent during an assessment to the specific endpoint. Without a strong binding of the endpoint to the posture information, an attacker could record the attributes sent during an assessment of a compliant endpoint and later replay those attributes so that a non-compliant endpoint can now gain access to the network or protected resource. This attack could be employed by a network MitM that is able to eavesdrop and proxy message exchanges, or by using local rogue agents on the endpoints. Assessments lacking some form of freshness exchange could be subject to replay of prior assessment data, even if it no longer reflects the current state of the endpoint. Use of a PT protocol providing strong integrity protection and authentication including a freshness exchange is necessary countermeasure to these attacks.

5.2.5. Attribute Insertion

Similar to the attribute modification attacks, an adversary wishing to include one or more attributes or PA-TNC messages inside a valid assessment may be able to insert the attributes or messages without detection by the recipient. For example, an attacker could add attributes to the front of a PA-TNC message to cause an assessment to succeed even for a non-compliant endpoint, particularly if it knew that the recipient ignored repeated attributes within a message. Similarly, if a Posture Collector or Posture Validator always generated an error if it saw unexpected attributes, the attacker could cause failures and denial of service by adding attributes or messages to an exchange. Use of a PT protocol providing strong authentication and integrity protection could prevent the adversary from inserting attributes into the assessment.

5.2.6. Denial of Service

A variety of types of denial-of-service attacks are possible against the PA-TNC message exchange if left unprotected from untrusted parties along the communication path between the Posture Collector and Posture Validator. Normally, the PT exchange is bidirectionally authenticated, which helps to prevent a MitM on the network from becoming an active proxy, but transparent message routing gateways may still exist on the communication path and can modify the integrity of the message exchange unless adequate integrity protection is provided. If the MitM or other entities on the network can send messages to the Posture Broker Client or Posture Broker Server that appear to be part of an assessment, these messages could confuse the Posture Collector and Posture Validator or cause them to perform unnecessary work or take incorrect action. Several example denial-of-service situations are described in sections 5.2.3 and 5.2.5. Many potential denial-of-service examples exist, including flooding messages to the Posture Collector or Posture Validator, sending very large messages containing many attributes, and repeatedly asking for resource-intensive operations.

6. Privacy Considerations

The PA-TNC protocol is designed to allow for controlled disclosure of security-relevant information about an endpoint, specifically for the purpose of enabling an assessment of the endpoint’s compliance with network policy. The purpose of this protocol is to provide visibility into the state of the protective mechanisms on the endpoint, in order for the Posture Validators and Posture Broker Server to determine whether the endpoint is up to date and thus has the best chance of being resilient in the face of malware threats. One risk associated with providing visibility into the contents of an endpoint is the increased chance for exposure of privacy-sensitive information without the consent of the user.

While this protocol does provide the Posture Validator the ability to request specific information about the endpoint, the protocol is not open ended, bounding the Posture Validator to only query specific information (attributes) about specific security features (component types) of the endpoint. Each PA-TNC message is explicitly about a single component from the list of components in section 3.5. These components include a list of security-related aspects of the endpoint that affect the ability of the endpoint to resist attacks and thus are of interest during an assessment. Discretionary components used by the user to create or view content are not on the list, as they are more likely to have access to privacy-sensitive information.

Similarly, PA-TNC messages contain a set of attributes that describe the particular component. Each attribute contains generic information (e.g., product information or versions) about the component, so it is unlikely to include any user-specific or identifying information. This combination of a limited set of security-related components with non-user-specific attributes greatly reduces the risk of exposure of privacy-sensitive information. Vendors that choose to define additional component types and/or attributes within their namespace are encouraged to provide similar constraints.

Even with the bounding of standard attribute information to specific components, it is possible that individuals might wish to share less information with different networks they wish to access. For example, a user may wish to share more information when connecting to or being reassessed by the user's employer network than what would be made available to the local coffee shop wireless network. While these situations do not impact the protocol itself, they do suggest that Posture Collector implementations should consider supporting a privacy filter allowing the user and/or system owner to restrict access to certain attributes based upon the target network.

The underlying PT protocol authenticates the network's Posture Broker Server at the start of an assessment, so identity can be made available to the Posture Collector and per-network privacy filtering is possible. Network owners should make available a list of the attributes they require to perform an assessment and any privacy policy they enforce when handling the data. Users wishing to use a more restricted privacy filter on the endpoint may risk not being able to pass an assessment and thus not gain access to the requested network or resource.

7. IANA Considerations

This section defines the contents of three new IANA registries: PA-TNC Attribute Types, PA-TNC Error Codes, and PA-TNC Remediation Parameters Types. This section explains how these registries work. Also, this specification defines several new PA Subtypes for use with PA-TNC.

All of the registries defined in this document support IETF standard values and vendor-defined values. To explain this phenomenon, we will use the PA-TNC Attribute Type as an example, but the other three registries work the same way. Whenever a PA-TNC Attribute Type appears on a network, it is always accompanied by an SMI Private Enterprise Number (PEN), also known as a vendor ID. If this vendor ID is zero, the accompanying PA-TNC Attribute Type is an IETF standard value listed in the IANA registry for PA-TNC Attribute

Types, and its meaning is defined in the specification listed for that PA-TNC Attribute Type in that registry. If the vendor ID is not zero, the meaning of the PA-TNC Attribute Type is defined by the vendor identified by the vendor ID (as listed in the IANA registry for SMI PENs). The identified vendor is encouraged but not required to register with IANA some or all of the PA-TNC Attribute Types used with their vendor ID and publish a specification for each of these values.

This delegation of namespace is analogous to the technique used for OIDs. It can result in interoperability problems if vendors require support for particular vendor-specific values. However, such behavior is explicitly prohibited by this specification (in section 4.1), which dictates that "Posture Collectors and Posture Validators MUST NOT require support for particular vendor-specific PA-TNC Attribute Types and MUST interoperate with other parties despite any differences in the set of vendor-specific PA-TNC Attribute Types supported (although they MAY permit administrators to configure them to require support for specific PA-TNC Attribute Types)". Similar requirements are included for PA Subtypes, Remediation Parameters Types, and PA-TNC Error Codes.

7.1. Designated Expert Guidelines

For all of the IANA registries defined by this specification, new values are added to the registry by Expert Review with Specification Required, using the Designated Expert process defined in RFC 5226 [3].

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

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

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

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

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

Section 7.2 defines the new PA Subtypes. The following three sections provide guidance to the IANA in creating and managing the new IANA registries defined by this specification.

7.2. PA Subtypes

Section 3.5 of this specification defines several new PA Subtypes that have been added to the PA Subtypes registry defined in the PB-TNC specification. Here is a list of these assignments:

PEN	Integer	Name	Defining Specification
0	0	Testing	RFC 5792
0	1	Operating System	RFC 5792
0	2	Anti-Virus	RFC 5792
0	3	Anti-Spyware	RFC 5792
0	4	Anti-Malware	RFC 5792
0	5	Firewall	RFC 5792
0	6	IDPS	RFC 5792
0	7	VPN	RFC 5792
0	8	NEA Client	RFC 5792

These PA Subtypes have been added to the registry for PA Subtypes defined in the PB-TNC specification, with this RFC as the reference.

7.3. Registry for PA-TNC Attribute Types

The name for this registry is "PA-TNC Attribute Types". Each entry in this registry should include a human-readable name, an SMI Private Enterprise Number, a decimal integer value between 0 and 2^32-1, and a reference to the specification where the contents of this attribute type are defined. This specification must define the meaning of this PA-TNC attribute type and the format and semantics of the PA-TNC Attribute Value field for PA-TNC attributes that include the designated Private Enterprise Number in the PA-TNC Attribute Vendor ID field and the designated numeric value in the PA-TNC Attribute Type field.

The following entries for this registry are defined in this document. They are the initial entries in the registry for PA-TNC Attribute Types. Additional entries to this registry are added by Expert Review with Specification Required, following the guidelines in section 7.1.

PEN	Integer	Name	Defining Specification
0	0	Testing	RFC 5792
0	1	Attribute Request	RFC 5792
0	2	Product Information	RFC 5792
0	3	Numeric Version	RFC 5792
0	4	String Version	RFC 5792
0	5	Operational Status	RFC 5792
0	6	Port Filter	RFC 5792
0	7	Installed Packages	RFC 5792
0	8	PA-TNC Error	RFC 5792
0	9	Assessment Result	RFC 5792
0	10	Remediation Instructions	RFC 5792
0	11	Forwarding Enabled	RFC 5792
0	12	Factory Default Password Enabled	RFC 5792
0	0xffffffff	Reserved	RFC 5792

7.4. Registry for PA-TNC Error Codes

The name for this registry is "PA-TNC Error Codes". Each entry in this registry should include a human-readable name, an SMI Private Enterprise Number, a decimal integer value between 0 and 2^32-1, and a reference to the specification where this error code is defined. This specification must define the meaning of this error code and the format and semantics of the Error Information field for PA-TNC attributes that have a PA-TNC vendor ID of 0, a PA-TNC Attribute Type of PA-TNC Error, the designated Private Enterprise Number in the PA-TNC Error Code Vendor ID field, and the designated numeric value in the PA-TNC Error Code field.

The following entries for this registry are defined in this document. They are the initial entries in the registry for PA-TNC Error Codes. Additional entries to this registry are added by Expert Review with Specification Required, following the guidelines in section 7.1.

PEN	Integer	Name	Defining Specification
0	0	Reserved	RFC 5792
0	1	Invalid Parameter	RFC 5792
0	2	Version Not Supported	RFC 5792
0	3	Attribute Type Not Supported	RFC 5792

7.5. Registry for PA-TNC Remediation Parameters Types

The name for this registry is "PA-TNC Remediation Parameters Types". Each entry in this registry should include a human-readable name, an SMI Private Enterprise Number, a decimal integer value between 1 and 2^32-1, and a reference to the specification where the contents of this remediation parameters type are defined. This specification must define the meaning of this PA-TNC Remediation Parameters Type and the format and semantics of the Remediation Parameters field for PA-TNC attributes that include the designated Private Enterprise Number in the Remediation Parameters Vendor ID field and the designated numeric value in the Remediation Parameters Type field.

The following entries for this registry are defined in this document. They are the initial entries in the registry for PA-TNC Remediation Parameters Types. Additional entries to this registry are added by Expert Review with Specification Required, following the guidelines in section 7.1.

PEN	Integer	Name	Defining Specification
0	0	Reserved	RFC 5792
0	1	URI	RFC 5792
0	2	Remediation String	RFC 5792

8. Acknowledgments

Thanks to the Trusted Computing Group for contributing the initial text [8] upon which this document was based. The authors would also like to acknowledge the following people who have contributed to or provided substantial input on the preparation of this document or predecessors to it: Stuart Bailey, Roger Chickering, Lauren Giroux, Charles Goldberg, Steve Hanna, Ryan Hurst, Meenakshi Kaushik, Greg Kazmierczak, Scott Kelly, PJ Kirner, Houcheng Lee, Lisa Lorenzin, Mahalingam Mani, Sung Lee, Ravi Sahita, Mauricio Sanchez, Brad Upson, and Han Yin.

9. References

9.1. Normative References

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

[2] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[3] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[4] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[5] 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.

[6] Phillips, A., Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

[7] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

9.2. Informative References

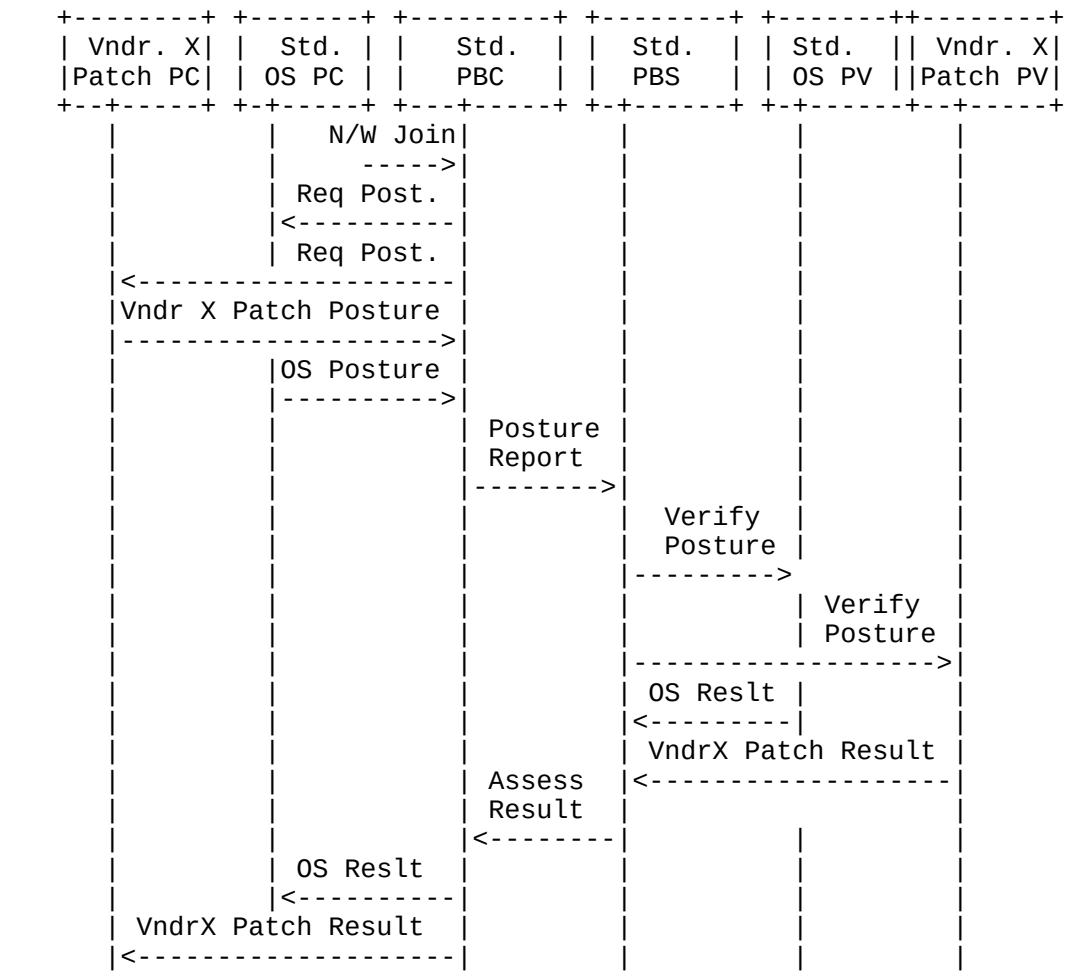
[8] Trusted Computing Group, "IF-M: TLV Binding", http://www.trustedcomputinggroup.org/resources/tnc_ifm_tlv_binding_specification, February 2010.

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

Appendix A. Use Cases

A.1. Initial Client-Triggered Assessment

This scenario involves the assessment of an endpoint initiated during network join. The assessment is triggered by the Posture Broker Client (PBC) and involves collection of patch information from both Standard Operating System (OS) Posture Collector and vendor-specific Patch Posture Collector (PC). The assessment by both the vendor-specific Patch Posture Validator (PV) and Standard OS Posture Validator result in a compliant assessment decision that results in a compliant System Assessment Decision to be returned by the Posture Broker Server (PBS).



A.1.1. Message Contents

This section shows the contents of the key fields in each of the PA messages exchanged in this use case. When necessary, additional commentary is provided to explain why certain fields contain the shown values. Note that many of the flows shown are between components on the same system so no message contents are shown.

A.1.1.1. N/W Join

This flow represents the event that causes the PBC to decide to start an assessment of the endpoint in order to gain access to the network. This is merely an event and does not include a message being sent.

A.1.1.2. Request Posture (Req Post.)

This flow illustrates an invocation of the OS and patch posture collectors requesting particular posture attributes to be sent. Because this use case is triggered locally, the contents of this flow aren't specified by NEA.

A.1.1.3. Vendor X Patch Posture (VndrX Patch Posture)

This flow contains the PA message from the Patch Posture Collector:

```
Vendor X Patch Posture PA Message  {
   Attribute HDR {Message ID}
   Attribute 1 {
      vendor-id=1 (vendor X)
      type=1 (Vendor X namespace attribute)
      length
      Value = {
         VendorXAttribute1=123
      }
   }
   Attribute 2 {
      vendor-id=1 (vendor X)
      type=2 (Vendor X namespace attribute)
      length
      Value = {
         VendorXAttribute2=456
      }
   }
}
```

A.1.1.4. OS Posture

This flow contains the PA message from the OS Posture Collector:

```
OS Posture PA Message  {

   Attribute HDR {Message ID}
   Attribute 1 {
      vendor-id=0
      type=2 (product information)
      length
      Value = {
         Product-vendor-id=311   -- Microsoft's PEN
         Product-name="Windows Vista"
      }
   }
   Attribute 2 {
      vendor-id=0
      type=3 (numeric version)
      length
      Value = {
         major-version=6     -- Vista is version 6.0
         minor-version=0
         build-number=456789
         service-pack-major=0   -- No service packs
         service-pack-minor=0
      }
   }
}
```

A.1.1.5. Posture Report

This flow contains the PB message containing the PA messages from the Patch and OS Posture Collectors; the message content is described in the PB-TNC specification.

A.1.1.6. Verify Posture

This flow illustrates an invocation of the OS and patch Posture Validators requesting verification of the posture attributes received. Because this flow happens locally within the NEA server, NEA does not specify the message contents.

A.1.1.7. OS Posture Result (OS Reslt)

This flow contains the PA message (Posture Assessment Result) from the OS Posture Validator

```
OS Posture Result PA Message {
   Attribute HDR {Message ID}
      Attribute 1 {
           vendor-id=0
           type=9 (assessment-result)
           length
           Value = {
              assessment-result=0 (compliant)
           }
      }
 }
```

A.1.1.8. Vendor X Patch Result (VndrX Patch Result)

This flow contains the PA message (Posture Assessment Result) from the Vendor X Patch Posture Validator

```
Patch Vendor X Posture Result PA Message {
   Attribute HDR {Message ID}
      Attribute 1 {
           vendor-id=0
           type=9 (assessment-result)
           length
           Value = {
              assessment-result=0 (compliant)
           }
      }
 }
```

A.1.1.9. Assessment Result (Assess Result)

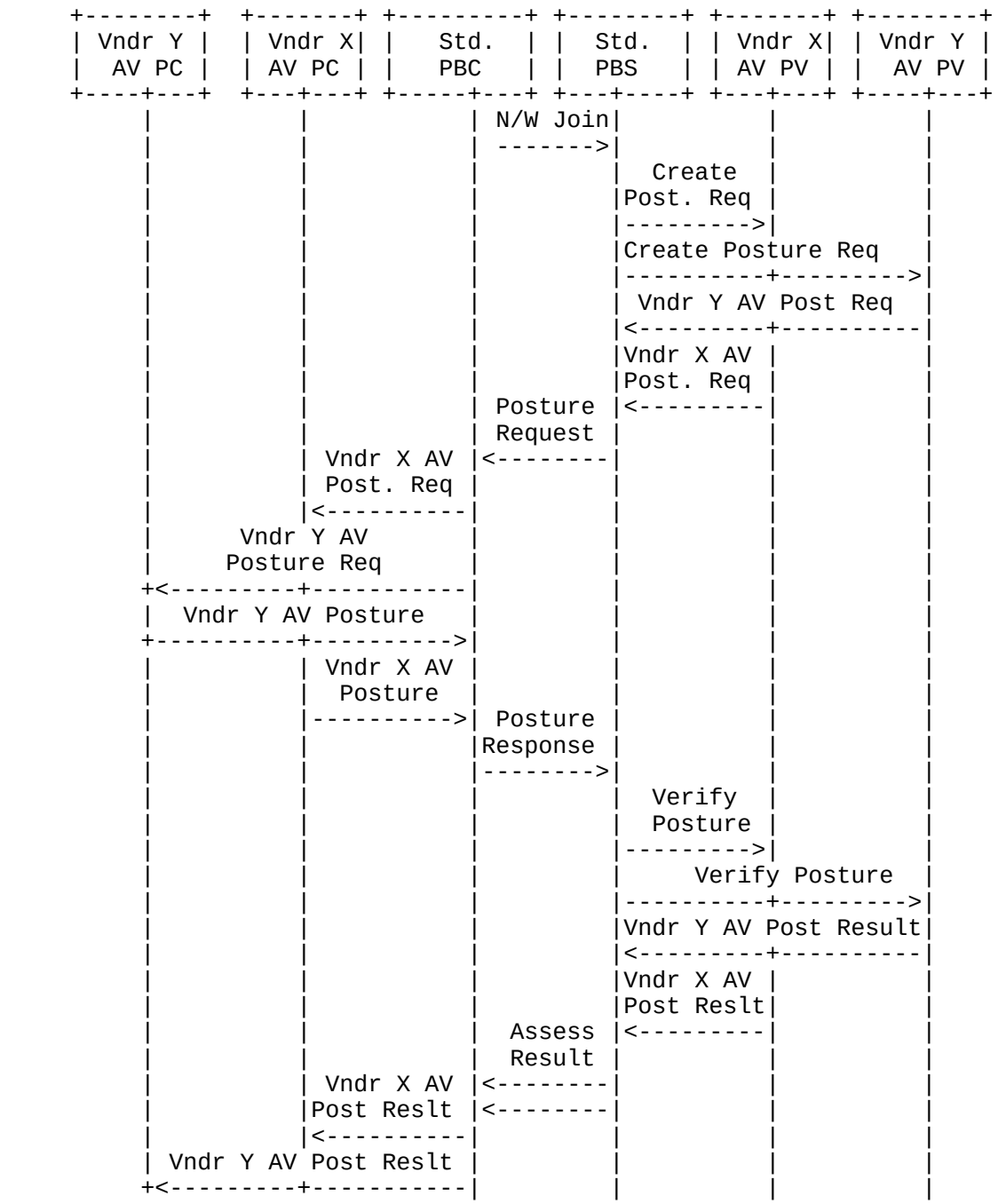
This flow contains the PB message containing the system assessment result computed by the Posture Broker Server and the PA messages from the Patch and OS Posture Validators; the message content is described in the PB-TNC specification.

A.1.1.10. Posture Result (OS PRslt & Vndr X Post PResult)

These flows illustrate an invocation of the OS and Vendor X Patch Posture Collectors to receive the posture assessment results. Because this flow is triggered locally, NEA does not specify the contents of this flow.

A.2. Server-Initiated Assessment with Remediation

This scenario involves the assessment of an endpoint initiated by the NEA Server. The assessment is triggered by the Posture Broker Server and involves collection of Anti-Virus attributes for two Anti-Virus components running on the endpoint. The endpoint is assessed to be compliant by one of the vendor (Vendor X) anti-virus Posture Validators and non-compliant by the other vendor (Vendor Y) anti-virus Posture Validator. Based upon the Posture Broker Server's policy, this results in a non-compliant system assessment decision to be returned by the Posture Broker Server. The Posture Broker Server also returns remediation instructions for the endpoint as part of the response.


Vndr Y AV PC
Vndr X AV PC
Std. PBC
Std. PBS
Vndr X AV PV
Vndr Y AV PV
N/W Join
Create Post. Req
Create Posture Req
Vndr Y AV Post Req
Vndr X AV Post. Req
Posture Request
Vndr X AV Post. Req
Vndr Y AV Posture Req
Vndr Y AV Posture
Vndr X AV Posture
Posture Response
Verify Posture
Verify Posture
Vndr Y AV Post Result
Vndr X AV Post Reslt
Assess Result
Vndr X AV Post Reslt
Vndr Y AV Post Reslt

A.2.1. Message Contents

This section shows the contents of the key fields in each of the PA messages exchanged in this use case. When necessary, additional commentary is provided to explain why certain fields contain the shown values. Note that many of the flows shown are between components on the same system so no message contents are shown.

A.2.1.1. N/W Join

This flow represents the event that causes the PBS to decide to start an assessment of the endpoint in order to gain access to the network. This is merely an event and does not include a message being sent.

A.2.1.2. Create Posture Request (Create Posture Req)

This flow illustrates an invocation of the Vendor X and Vendor Y Anti-Virus Posture Validators enabling posture request attributes to be created. Because this use case is triggered locally, NEA does not specify the contents of this flow.

A.2.1.3. Vendor Y AV Posture Request (Vndr Y AV Posture Req)

This flow contains the PA message (Posture Request) from the Vendor Y Anti-Virus Posture Validator

```
Vendor Y AV Posture Request PA Message {
    Attribute HDR {Message ID}
       Attribute 1 {
           vendor-id=0
           type=1 (Attribute Request)
           length
           Value = {
              Vendor-id=0 (IETF Standard)
              Type=2 (Standard attribute, Product-Information)
              Vendor-id=1 (Vendor Y)
              Type=2 (Vendor Y attribute, Extended-Dat-Version)
            }
       }
}
```

A.2.1.4. Vendor X AV Posture Request (Vndr X AV Post. Req)

This flow contains the PA message (Posture Request) from the Vendor X Anti-Virus Posture Validator

```
Vendor X AV Posture Request PA Message {
    Attribute HDR {Message ID}
       Attribute 1 {
           vendor-id=0
           type=1 (Attribute Request)
           length
           Value = {
              Vendor-id=1 (Vendor X)
              Type=1 (Vendor X attribute, Scan-Engine-Version)
              Vendor-id=0 (IETF Standard)
              Type=5 (Standard, Operational-Status)
           }
       }
 }
```

A.2.1.5. Posture Request

This flow contains the PB message containing the PA messages from the Vendor X and Vendor Y Anti-Virus Posture Validators; the message content is described in the PB-TNC specification.

A.2.1.6. Posture Request (Vndr X AV Post Req & Vndr Y AV Post Req)

These flows illustrate an invocation of the Vendor X and Vendor Y Anti-Virus Posture Collectors to process the Posture Request and return the particular posture attributes requested. Because this flow is triggered locally, NEA does not specify the contents of this flow.

A.2.1.7. Vendor Y AV Posture (Vndr Y AV Posture)

This flow contains the PA message (response to the Posture Request) from the Vendor Y Anti-Virus Posture Collector.

```
Vendor Y AV Posture PA Message {
  Attribute HDR {Message ID}
      Attribute 1 {
         vendor-id=0 (IETF Standard)
         Type=2 (Standard attribute, Product-Information)
         length
         Value = {
            product-vendor-id=12345 (vendor Y)
            product-id=987 (AV product id from vendor Y)
            product-name="Vendor Y Anti-Virus"
         }
      }
      Attribute 2 {
         vendor-id=2 (vendor Y)
         type=2 (vendor Y attribute, DAT-Version)
         length
         Value = {
            DAT-version=5678
         }
      }
  }
```

A.2.1.8. Vendor X AV Posture (Vndr X AV Posture)

This flow contains the PA message (response to the Posture Request) from the Vendor X Anti-Virus Posture Collector.

```
Vendor X AV Posture PA Message {
   Attribute HDR {Message ID}
      Attribute 1 {
         vendor-id=1
         type=1 (vendor X attribute, Scan-Engine-Version)
         length
         Value = {
            scan-engine-version=1234
         }
      }
      Attribute 2 {
         vendor-id=0 (IETF Standard)
         type=5 (Standard, Operational-Status)
         length
         Value = {
            status=2 (installed but non-operational)
            result=0 (unknown)
            last use="" (never used)
          }
      }
  }
```

A.2.1.9. Posture Response

This flow contains the PB message containing the PA messages from the Vendor X and Vendor Y Anti-Virus Posture Collectors; the message content is described in the PB-TNC specification.

A.2.1.10. Verify Posture

This flow illustrates an invocation of the Vendor X and Vendor Y Anti-Virus Posture Validators requesting verification of the posture attributes received. Because this flow happens locally within the NEA server, NEA does not specify the message contents.

A.2.1.11. Vendor Y AV Posture Result (Vndr Y AV Post Result)

This flow contains the PA message (Posture Assessment Result) from the Vendor Y Anti-Virus Posture Validator

```
Vendor Y AV Posture Result PA Message {
   Attribute HDR {Message ID}
     Attribute 1 {
        vendor-id=0
        type=9 (assessment-result)
        length
        Value = {
           assessment-result=0 (compliant)
        }
     }
  }
```

A.2.1.12. Vendor X AV Posture Result (Vndr X AV Post Reslt)

This flow contains the PA message (Posture Assessment Result) from the Vendor X Anti-Virus Posture Validator

```
Vendor X AV Posture Result PA Message {
    Attribute HDR {Message ID}
      Attribute 1 {
         vendor-id=0
         type=9 (assessment-result)
         length
         Value = {
            assessment-result=1 (non-compliant)
         }
      }
 }
```

A.2.1.13. Assessment Result (Assess Result)

This flow contains the PB message containing the system assessment result computed by the Posture Broker Server and the PA messages from the Vendor X and Vendor Y Anti-Virus Posture Validators; the message content is described in the PB-TNC specification.

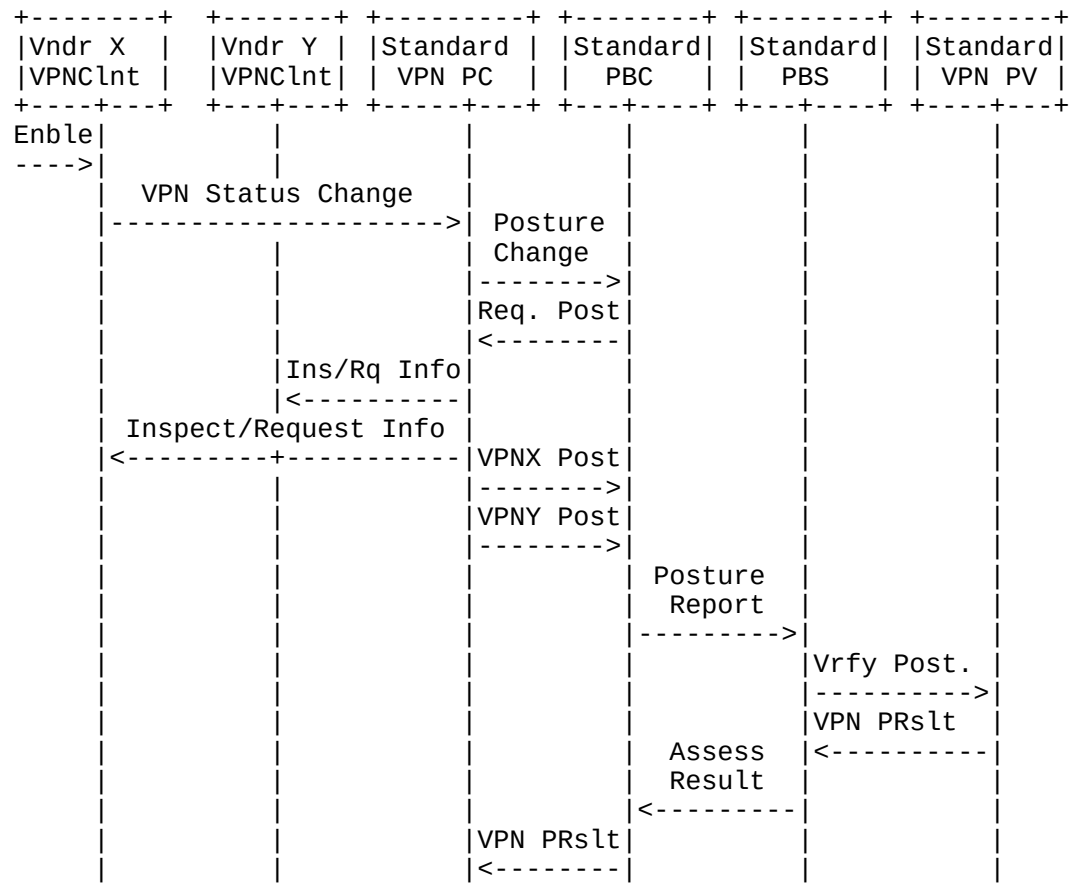
A.2.1.14. Posture Result (Vndr X AV Post Reslt & Vndr Y AV Post Reslt)

These flows illustrate an invocation of the Vendor X and Vendor Y Anti-Virus Posture Collectors to receive the posture assessment results. Because this flow is triggered locally, NEA does not specify the contents of this flow.

A.3. Client-Triggered Reassessment

This scenario involves the reassessment of an endpoint as a result of enabling a software component on the endpoint. The endpoint has two VPN client software components, one from vendor X for the user's home network and other from vendor Y for the network that the endpoint is currently accessing. The assessment is triggered when the user tries to use the Vendor X VPN client; this is a violation of the assessment policy. The Posture Broker Client triggers the posture assessment when it receives a notification from the VPN Posture Collector about the change to the operational state of the VPN component on the endpoint. Note that the VPN Posture Collector may support standard attributes and some vendor-defined attributes from vendor X's and vendor Y's namespaces. This use case does not leverage vendor-defined attributes. The assessment involves verification of the standard VPN posture attributes by the standard VPN Posture Validator that results in a non-compliant assessment result.

This use case relies on the use of multiple Posture Collector IDs for a single Posture Collector as described in section 3.3 of the PA-TNC specification. In this example, the Posture Collector will obtain two Posture Collector IDs to a single Posture Collector (Standard VPN PC) and the Posture Collector will generate two separate PA messages each using a different ID to report the posture for Vendor X and Vendor Y VPN Clients. The Posture Broker Client will associate the assigned IDs in the PB message sent to the NEA Server. This entire behavior will be completely opaque to the NEA Server, which will handle the PB message as if there were two VPN Posture Collectors on the NEA Client.



A.3.1. Message Contents

This section shows the contents of the key fields in each of the PA messages exchanged in this use case. When necessary, additional commentary is provided to explain why certain fields contain the shown values. Note that many of the flows shown are between components on the same system so no message contents are shown.

A.3.1.1. Enable VPN Client (Enble)

This flow represents the end user triggered event of starting the VPN Client software from Vendor X. This is merely an event and does not include a message being sent.

A.3.1.2. Notify Status Change (VPN Status Change)

This flow represents the detection of the active state of the Vendor X VPN Client software by the VPN Posture Collector. This is merely an event and does not include a message being sent.

A.3.1.3. Notify Posture Change (Posture Change)

This flow represents the notification of the VPN posture change sent from the VPN Posture Collector to the Standard Posture Broker Client. This is merely an event and does not include a message being sent.

A.3.1.4. Request Posture (Req. Post)

This flow illustrates an invocation of the VPN Posture Collector requesting particular posture attributes to be sent. Because this use case is triggered locally, NEA does not specify the contents of this flow.

A.3.1.5. Inspect/Request Info (Ins/Rq Info)

This flow illustrates the acquisition of the posture information by the VPN Posture Collector from the Vendor X and Vendor Y VPN Client components. Because this flow is triggered locally, NEA does not specify the message contents.

A.3.1.6. Vendor X VPN Posture (VPNX Post)

This flow contains the PA message from the VPN Posture Collector describing the Vendor X VPN Client's posture:

```
Vendor X VPN Posture PA Message{
   Attribute HDR {Message ID}
     Attribute 1 {
           vendor-id=0
           type=2 (product information)
           length
           Value = {
              product-vendor-id=9876 (vendor X)
              product-id=567 (VPN client identifier for Vndr X)
              product-name="Vendor X VPN Client"
            }
      }
      Attribute 2 {
           vendor-id=0
           type=5 (operational status)
           length
           Value = {
              Status=3 (Operational)
              Result=1 (Successful use with no errors detected)
              last Use="2008-07-07T12:00:00Z"
           }
      }
```

A.3.1.7. Vendor Y VPN Posture (VPNY Post)

This flow contains the PA message from the VPN Posture Collector including the Vendor Y VPN Client's posture:

```
Vendor Y VPN Posture PA Message{
   Attribute HDR {Message ID}
       Attribute 1 {
           vendor-id=0
           type=2 (product information)
           length
           Value = {
              product-vendor-id=Vendor Y
              product-id=234 (VPN client identifier for Vndr Y)
              product-name="Vendor Y VPN Client"
            }
       }
       Attribute 2 {
           vendor-id=0
           type=5 (operational status)
           length
           Value = {
             Status=3 (Operational)
             Result=1 (Successful use with no errors detected)
             last Use="2008-07-07T14:05:00Z"
           }
       }
}
```

A.3.1.8. Posture Report

This flow contains the PB message containing the PA message from the VPN Posture Collector; the message content is described in the PB-TNC specification.

A.3.1.9. Verify Posture (Vrfy Post.)

This flow illustrates an invocation of the VPN Posture Validator requesting verification of the posture attributes received. Because this flow happens locally within the NEA Server, NEA does not specify the message contents.

A.3.1.10. VPN Posture Result (VPN PRslt)

This flow contains the PA message (Posture Assessment Result) from the VPN Posture Validator

```
VPN Posture Result PA Message {
   Attribute HDR {Message ID}
      Attribute 1 {
           vendor-id=0
           type=9 (assessment-result)
           length
           Value = {
              assessment-result=1 (non-compliant)
           }
      }
 }
```

A.3.1.11. Assessment Result (Assess Result)

This flow contains the PB message containing the system assessment result computed by the Posture Broker Server and the PA messages from the VPN Posture Validator; the message content is described in the PB-TNC specification.

A.3.1.12. Posture Result (VPN PRslt)

This flow illustrates an invocation of the VPN Posture Collector to receive the posture assessment result. Because this flow is triggered locally, NEA does not specify the contents of this flow.

Appendix B. Evaluation against NEA Requirements

This section evaluates the PA-TNC protocol against the requirements defined in the NEA Requirements document. Each subsection considers a separate requirement from the NEA Requirements document. Only common requirements (C-1 through C-10) and PA requirements (PA-1 through PA-6) are considered, since these are the only ones that apply to PA.

B.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.

PA-TNC meets this requirement. It allows an unlimited number of round trips between the NEA Client and NEA Server.

B.2. Evaluation against Requirement 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.

PA-TNC meets this requirement. PA-TNC is designed to work whether the NEA Client or the NEA Server initiates a posture assessment or reassessment.

B.3. Evaluation against Requirement 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.

Security for PA-TNC messages being sent over the network is provided through PT protocol security. Therefore, PA-TNC does not include any security capabilities. Since this requirement only applies to NEA protocols "including security capabilities", this specification is not subject to this requirement (see section 5.2).

B.4. Evaluation against Requirement 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).

PA-TNC meets this requirement. PA-TNC can operate over any PT protocol that meets the requirements for PT stated in the NEA Requirements document. PA-TNC does not have any dependencies on specific details of the underlying PT protocol.

B.5. Evaluation against Requirement 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, PA-TNC should receive a strong preference. PA-TNC is equivalent with IF-M 1.0, an open TCG specification. Other specifications from TCG and other groups are also under development based on the IF-M 1.0 specification. Selecting PA-TNC as the basis for the PA protocol will ensure compatibility with IF-M 1.0, with these other specifications, and with their implementations.

B.6. Evaluation against Requirement 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.

PA-TNC meets this requirement. PA-TNC supports an unlimited number of Posture Collectors, Posture Validators, NEA Clients, and NEA Servers. It also is quite scalable in many other aspects as well. A PA-TNC message can contain up to 2^32-1 octets and about 2^28 PA-TNC attributes. Each organization with an SMI Private Enterprise Number is entitled to define up to 2^32 vendor-specific PA-TNC Attribute Types, 2^16 vendor-specific PA-TNC Product IDs, and 2^32 vendor-

specific PA-TNC Error Codes. Each attribute can contain almost 2^32 octets. It is generally not advisable or necessary to send this much data in a NEA assessment, but still PA-TNC is highly scalable and meets requirement C-6 easily.

B.7. Evaluation against Requirement 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.

PA-TNC meets this requirement. Each PA-TNC message can contain about 2^28 PA-TNC attributes. PA-TNC supports up to 2^32 round trips in a session so the maximum number of attribute messages that can be sent in a single session is actually about 2^50. However, it is generally inadvisable and unnecessary to send a large number of messages in a NEA assessment. As for efficiency, PA-TNC adds only 12 octets of overhead per attribute and 8 octets per message (which is negligible on a per-attribute basis).

B.8. Evaluation against Requirement C-8

Requirement C-8 says:

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

PA-TNC meets this requirement. A PA-TNC exchange is envisioned (based on current deployment experience) to involve one or two round trips with less than 500 octets of PA-TNC messages. Of course, use of vendor-specific PA-TNC attribute types could expand the assessment. However, PA-TNC itself imposes an overhead of only 8 octets per PA-TNC message and 12 octets per attribute.

B.9. Evaluation against Requirement 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.

PA-TNC meets this requirement. The only field included in a PB-TNC attribute for display to the user includes a language tag that could be selected based upon the user’s PB-TNC negotiated preferred language for the assessment (see section 4.10 of the PB-TNC

specification). With this exception, all of the strings in the standard PA-TNC attributes are intended for logging and programmatic comparisons.

If any vendor-specific PA-TNC attribute types or future IETF Standard PA-TNC Attribute Types include strings that are intended for display to a user, they should be translated to the user's preferred language. The Posture Broker Server will need to expose the user's preferences to the Posture Validators through whatever API or protocol is used to connect those components. However, that is all out of scope for this specification.

B.10. Evaluation against Requirement C-10

Requirement C-10 says:

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

PA-TNC meets this requirement. All strings in the PA-TNC protocol are encoded in UTF-8 format. This allows the protocol to support a wide range of languages efficiently.

B.11. Evaluation against Requirement 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 round trips, 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.

PA-TNC meets this requirement. The design of the PA-TNC protocol emphasizes efficient transport of information in order to maximize its usability in constrained PT environments. Local APIs could allow Posture Collectors and Posture Validators to discover when they are operating in a less constrained deployment and then make use of more verbose attributes. Similarly, Posture Collectors could choose not to send or use smaller attributes (including assertions from previous assessments) when faced with a very constrained network connection.

B.12. Evaluation against Requirement PA-1

Requirement PA-1 says:

PA-1 The PA protocol MUST support communication of an extensible set of NEA standards-defined attributes. These attributes will be uniquely identifiable from non-standard attributes.

PA-TNC meets this requirement. Each attribute is identified with a PA-TNC Attribute Vendor ID and a PA-TNC Attribute Type. IETF Standard PA-TNC Attribute Types use a vendor ID of zero (0), in contrast with vendor-specific PA-TNC Attribute Types, which will use the vendor's SMI Private Enterprise Number as the vendor ID. The IANA will maintain a registry of PA-TNC Attribute Types with new values added by Expert Review with Specification Required, as described in the IANA Considerations section of this specification. Thus, the set of standard attribute types is extensible, but all standard attribute types are uniquely identifiable.

B.13. Evaluation against Requirement PA-2

Requirement PA-2 says:

PA-2 The PA protocol MUST support communication of an extensible set of vendor-specific attributes. These attributes will be segmented into uniquely identifiable vendor-specific namespaces.

PA-TNC meets this requirement. Each attribute is identified with a PA-TNC Attribute Vendor ID and a PA-TNC Attribute Type. Vendor-defined PA-TNC Attribute Types use the vendor's SMI Private Enterprise Number as the PA-TNC Attribute Vendor ID. Each vendor can define up to 2^32 PA-TNC Attribute Types, using its own internal processes to manage its set of attribute types.

The IANA is not involved, other than the initial assignment of the vendor's SMI Private Enterprise Number. Thus, the set of vendor-specific attributes is segmented into uniquely identifiable vendor-specific namespaces.

B.14. Evaluation against Requirement PA-3

Requirement PA-3 says:

PA-3 The PA protocol MUST enable a Posture Validator to make one or more requests for attributes from a Posture Collector within a single assessment. This enables the Posture Validator to reassess the posture of a particular endpoint feature or to request additional posture including from other parts of the endpoint.

PA-TNC meets this requirement. The Attribute Request attribute type is an IETF Standard PA-TNC Attribute Type that permits a Posture Validator to send to one or more Posture Collectors a request for one or more attributes. This attribute may be sent at any point in the posture assessment process and may in fact be sent more than once if the Posture Validator needs to first determine the type of operating system and then request certain attributes specific to that operating system, for example.

B.15. Evaluation against Requirement PA-4

Requirement PA-4 says:

PA-4 The PA protocol MUST be capable of returning attributes from a Posture Validator to a Posture Collector. For example, this might enable the Posture Collector to learn the specific reason for a failed assessment and to aid in remediation and notification of the system owner.

PA-TNC meets this requirement. A Posture Validator can easily send attributes to one or more Posture Collectors.

B.16. Evaluation against Requirement PA-5

Requirement PA-5 says:

PA-5 The PA protocol SHOULD provide authentication, integrity, and confidentiality of attributes communicated between a Posture Collector and Posture Validator. This enables end-to-end security across a NEA deployment that might involve traversal of several systems or trust boundaries.

PA-TNC does not include an explicit PA-level security mechanism but does lay a foundation allowing attribute-level security protections to be added later. As an existence proof, the NEA working group considered an Internet-Draft proposal capable of encapsulating PA attributes within a Cryptographic Message Syntax (CMS) security wrapper in a new attribute type. This proposal offered the protections described in this requirement. However, the NEA WG decided that the use cases in scope for the working group did not require PA-level security. The use cases involving PA message traversal of multiple systems or trust boundaries were considered out of scope; therefore, a Posture Validator to Posture Collector end-to-end security protection was considered not to be required.

Instead, PA-TNC attributes are protected by the PT layer authentication, integrity, and confidentiality support. This protects the attributes communicated between the Posture Transport

Client and Posture Transport Server. Because the Posture Collector is in the same address space as the Posture Broker Client and Posture Transport Client and the Posture Validator is in the same address space as the Posture Broker Server and Posture Transport Server, the underlying broker and transport components are deemed trusted with respect to not tampering with the PA messages (see trust model in section 5.1 for details). Encrypting the PA-TNC messages would not prevent a hostile broker or transport component from attacking the messages.

B.17. Evaluation against Requirement PA-6

Requirement PA-6 says:

PA-6 The PA protocol MUST be capable of carrying attributes that contain non-binary and binary data including encrypted content.

PA-TNC meets this requirement. PA-TNC attributes can contain non-binary and binary data including encrypted content. For examples, see the attribute type definitions contained in this specification.

Authors' Addresses

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

Kaushik Narayan
Cisco Systems Inc.
10 West Tasman Drive
San Jose, CA 95134
USA
EMail: kaushik@cisco.com