TCG Trusted Network Communications
TNC IF-M: TLV Binding

Specification Version 1.0
Revision 41
8 May 2014
Published

Contact:
admin@trustedcomputinggroup.org

TCG

TCG Published

Disclaimers, Notices, and License Terms

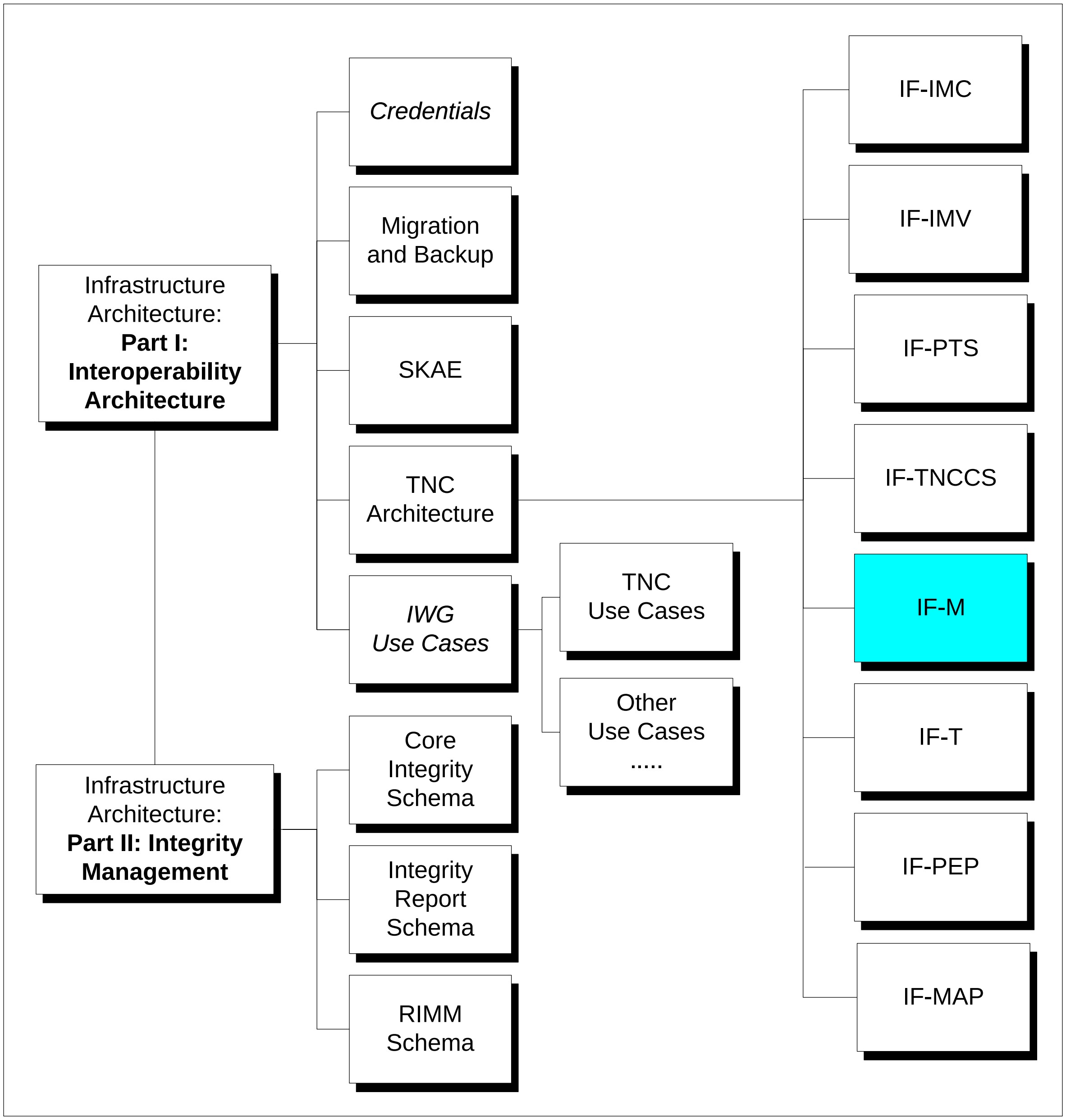
THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

IWG TNC Document Roadmap



Acknowledgement

The TCG wishes to thank all those who contributed to this specification. This document builds on considerable work done in the various working groups in the TCG.

Special thanks to the members of the TNC contributing to this document:

Scott Kelly	**Aruba Networks**
Mahalingam Mani	**Avaya**
Jeffery Dion	**The Boeing Company**
Steven Venema	**The Boeing Company**
Peter Wrobel	**CESG**
Nancy Cam-Winget	**Cisco Systems**
Scott Pope	**Cisco Systems**
Max Pritikin	**Cisco Systems**
Allan Thomson	**Cisco Systems**
Aaron Woland	**Cisco Systems**
Mark Townsend	**Enterasys**
Henk Birkholz	**Fraunhofer SIT**
Hidenobu Ito	**Fujitsu Limited**
Houcheng Lee	**Fujitsu Limited**
Sung Lee	**Fujitsu Limited**
Kazuaki Nimura	**Fujitsu Limited**
Mauricio Sanchez	**Hewlett-Packard**
Ira McDonald	**High North**
Han Yin	**Huawei**
Jiwei Wei	**Huawei**
Ren Lanfang	**Huawei**
Diana Arroyo	**IBM**
Emily Ratliff	**IBM**
Guha Prasad Venkataraman	**IBM**
Stuart Bailey	**Infoblox**
James Tan	**Infoblox**
Ravi Sahita	**Intel Corporation**
Ned Smith	**Intel Corporation**
Josh Howlett	**JANET**
Roger Chickering	**Juniper Networks**
Charles Goldberg	**Juniper Networks**
Steve Hanna (Editor)	**Juniper Networks**
Clifford Kahn	**Juniper Networks**
Lisa Lorenzin (TNC-WG co-chair)	**Juniper Networks**
Cliff Kahn	**Juniper Networks**
John Jerrim	**Lancope, Inc.**
Tom Price	**Lumeta**
Matt Webster	**Lumeta**
Kent Landfield	**McAfee**
Ryan Hurst	**Microsoft Corporation**
Atul Shah (TNC-WG co-chair)	**Microsoft Corporation**
Beth Abramowicz	**The MITRE Corporation**
Charles Schmidt	**The MITRE Corporation**
Sandilya Garimella	**Motorola**
Rainer Enders	**NCP**
Meenakshi Kaushik	**Nortel Networks**

Dick Wilkins	**Phoenix Technologies**
Paul Sangster	**Symantec**
Matthew Gast	**Trapeze Networks**
Greg Kazmierczak	**Wave Systems**
Thomas Hardjono	**Wave Systems**
Brad Upson	**UNH, Interoperability Lab**
Jessica Fitzgerald-McKay	**U.S. Government**
Richard Struse	**U.S. Government**
David Waltermire	**U.S. Government**

Table of Contents

1 Scope and Audience

The Trusted Network Communications Work Group (TNC-WG) has defined an open solution architecture that enables network operators to enforce policies regarding the security state of endpoints in order to determine whether to grant access to a requested network infrastructure. This security assessment of each endpoint is performed using a set of asserted integrity measurements covering aspects of the operational environment of the endpoint. Part of the TNC architecture is IF-M, a standard protocol between Integrity Measurement Collectors on the TNC Client to the Integrity Measurement Verifiers on the TNC Server. This document defines and specifies standard messages for the IF-M protocol.

Architects, designers, developers and technologists who wish to implement, use, or understand IF-M should read this document carefully. Before reading this document any further, the reader should review and understand the TNC architecture as described in [IF-ARCH].

1.1 Interoperable with IETF PA-TNC

One of the goals of the Trusted Network Communications WG is to maximize interoperability using open standards. As part of fulfilling this goal, the TNC WG chose to take the TCG standard IF-M protocol to the IETF for standardization. The initial version of IF-M was placed in "public review" status until the IETF standardization process had completed allowing both the TCG and IETF to publish interoperable standards at approximately the same time. The 1.0 version of this specification defines the IF-M protocol that is interoperable with PA-TNC [PA-TNC]. It is the current intention of the TNC WG to keep the IF-M and PA-TNC protocols interoperable for the future.

1.2 IETF Terminology Mapping to TNC

In case readers of this specification are also looking at the IETF Network Endpoint Assessment (NEA)'s PA-TNC specification, this section provides some guidance on how the terminology aligns between the IETF and NEA specifications.

PA-TNC - IETF NEA name for the application layer protocol that is interoperable with IF-M. "PA" is short for "Posture Attribute" protocol and "-TNC" highlights that the protocol is based upon work originally submitted by the TNC and is interoperable with this specification.

PB-TNC - IETF NEA name for the protocol between the NEA client to NEA server that is interoperable with the TNC's IF-TNCCS 2.0. Just as with the PA-TNC, the PB-TNC [PB-TNC] protocol is based upon work originally submitted by the TNC and is interoperable with IF-TNCCS 2.0 thus carries the "-TNC" suffix.

Posture – IETF NEA term for "measurement information" or "integrity measurement" used by TNC. The posture is returned from the NEA client (typically from its Posture Collectors) as part of an assessment. This is synonymous with the measurement information returned by the TNC client's IMCs.

Posture Collector – IETF NEA term synonymous with TNC's Integrity Measurement Collector (IMC)

Posture Validator – IETF NEA term synonymous with TNC's Integrity Measurement Validator (IMV)

2 Background

2.1 Purpose of IF-M

This document describes and specifies IF-M, an application level protocol capable of carrying Integrity Check Handshake messages between the Integrity Measurement Collectors (IMCs) on the TNC Client and the Integrity Measurement Verifiers (IMVs) on the TNC Server. IF-M is a multiple roundtrip messaging protocol that enables IMC(s) to send measurement information about local components on the endpoint to IMV(s) for evaluation against network security policy. The IMV(s) can respond using IF-M messages with additional requests for measurement data or optionally with its IMV Action Recommendation. The decision might also include a set of remediation instructions that the IMC could perform to bring its associated component into compliance with the IMV's policy.

The IF-M protocol is carried over the network by the IF-TNCCS protocol [IF-TNCCS][IF-TNCCS-SOH]. The TNC Client and Server pass the received IF-M messages to the appropriate set of IMCs and IMVs using the respective IF-IMC[IF-IMC] and IF-IMV[IF-IMV] APIs.

This specification defines the standard IF-M messages that can be used to enable interoperability between an IMC from one vendor and an IMV from another vendor. However, these messages are only a subset of the broader IF-M protocol. For years prior to the IF-M specification, IMC and IMV vendors have been employing vendor-defined IF-M messages to communicate between their IMCs and IMVs. It is expected that in the future a mixture of vendor-defined IF-M messages and standard IF-M messages may be employed to provide the best of both worlds: interoperability between IMCs and IMVs from different vendors and the tight integration that vendor-specific IF-M messages provide.

IF-M is a robust, extensible protocol capable of exchanging sets of attributes between zero or more subscribed IMCs and IMVs. This specification includes the definition of the message protocol and a partial set of standard attributes that are expected to have general applicability to different types of components (e.g. version information). IF-M attribute name space is extensible allowing for vendor-defined attributes to be established that convey product specific information and to allow for additional standard attributes to be defined in future specifications as more deployment experience becomes available.

2.2 Supported Use Cases

This section describes the IF-M use cases that must be supported. In order to focus on the protocol interactions, these use cases do not describe what triggered the assessment to occur. Thus it's expected that each use case is able to work regardless of what triggers the TNCC or TNCS to start the assessment. For example, such triggers include: an endpoint initially joining the network, a change to an endpoint or a significant event on the endpoint, the TNCS's policy change or the TNCS receiving notice of a significant event. Each type of trigger must be able to cause an assessment to occur even if a prior assessment of the endpoint has already occurred. In each case, the following use case message exchanges must be supported.

2.2.1 TNCC Initiated Assessment Use Case

1) TNCC determines that an assessment of the endpoint is required. This might occur if the TNCC becomes aware that the endpoint is trying to access an 802.1X protected network.

2) TNCC invokes one or more IMCs to send measurement information to their corresponding IMVs on the TNCS. Each IMC consults its policy and could choose to:

   a) Send nothing, if unwilling or unable to participate

   b) Send a hello message indicating its availability to respond to requests

   c) Send a subset of the measurements it's able to collect

   d) Send all of its collected measurements

   e) Send previously received and cached assertion attributes

3) TNCS passes attributes sent by IMCs to IMVs that have expressed an interest in the measurement information received from the TNCC. Each IMV consults its assessment policy and could choose to:
    a) Send nothing (e.g. might just be in monitoring or audit mode)
    b) Request additional measurement information from the IMC(s) by sending one or more IF-M messages. These IF-M messages could provide session specific information that should be bound into the measurement response (e.g. an IMV providing a nonce to be used during a TPM_Quote by the IMC). This is the expected behavior for each received hello message from an IMC. In response to the IMV requests for measurement information, the IMCs repeat step 1 above. This could lead to additional IMV requests in step 2.
    c) Make an assessment decision based on the measurements provided, return the IMV Evaluation Result to the TNCS and take one or more of the following actions:
        i) Send no IF-M messages
        ii) Send the component level IMV Action Recommendation to the IMCs
        iii) Send remediation instructions to the IMCs
        iv) Send assertion attributes to the IMCs. The assertion attributes could describe the level of compliance determined by the IMV.

2.2.2 TNCS Initiated Assessment Use Case

1) TNCS determines that an assessment of an endpoint is required. For instance, this could occur due to the TNCS becoming aware of an event occurring, due to a policy requiring periodic reassessment, or due to a TNCS policy change.
2) TNCS invokes one or more IMVs to initiate the assessment. Each invoked IMV could send requests for measurement information to their corresponding IMCs on the TNCC. Each invoked IMV consults its policy and could choose to:
    a) Send no IF-M messages, if unwilling or unable to participate
    b) Send a request for measurement information attributes. This information request could also include sending session specific information that should be bound into the measurement response (e.g. IMV providing a nonce to be used during a TPM_Quote by the IMC).
3) TNCC passes the attribute requests sent by IMVs to IMCs that have expressed a willingness to provide the requested measurement information. Each IMC consults its attribute policy and could choose to:
    a) Send no IF-M messages, if unwilling or unable to participate
    b) Send a subset of the requested measurement attributes that it’s able to collect (possibly factoring in privacy policy)
    c) Send all of the requested measurement attributes
    d) Send previously received and cached assertion attributes (possibly in addition to some requested measurement attributes)
4) TNCS passes attributes sent by IMCs to IMVs that have expressed an interest in the measurement information received from the TNCC. Each IMV consults its assessment policy and could choose to:
    a) Send no IF-M messages (e.g. might just be in monitoring or audit mode)
    b) Request additional measurement information from the IMC(s) by sending one or more IF-M messages. For example, this might occur if the initial roundtrip was to determine the platform type and operating system information and this next message will request specifics potentially unique to this endpoint. In response to the IMV requests for measurement information, the IMCs repeat step 3 above. This could lead to additional IMV requests in step 4.
    c) Make an assessment decision based on the measurements provided and then return the IMV Evaluation Result to the TNCS and take one or more of the following actions::
        i) Send no IF-M message

ii) Send the component level IMV Action Recommendation to the IMCs

iii) Send remediation instructions to the IMCs

iv) Send assertion attributes to the IMCs. The assertion attributes could describe the level of compliance determined by the IMV.

2.3 Non-supported Use Cases

This section describes the IF-M use cases that are considered out of scope for this specification.

2.3.1 IF-M Involving Remote IMV Use Case

This use case augments the TNCC and TNCS Initiated Assessment use cases above by moving the IMVs from being local to the TNCS and NAA to existing remotely on another system on the network. This use case highlights the potential security exposure of having IMVs remote to the NAA. Because these messages potentially contain security sensitive information (e.g. remediation instructions) they may require additional protection to when the TNC Server components are co-located on a single system.

In order to not repeat all of the other use cases with just a single alteration, this section bases its description on section 2.2.1 and highlights the significant differences using underlining. The following description assumes that at least one IMV is located remotely from the TNCS and no other security is provided on the link between the TNCS and IMV. An alternative solution to the remote IMV security concern is to use a secure protocol between the TNCS and the remote IMV. However, IF-M security might also be useful if the IMC or IMV does not trust the TNCC and TNCS to see unencrypted messages or if the IMV wishes only to accept information from a recognized authenticated IMC on the endpoint that is known to be reliable for both reporting measurements and performing remediation. This might not be the case for all IMC registered to receive message about a component on some endpoints.

1) TNCC invokes one or more IMCs to send measurement information to their corresponding IMVs on the TNCS. Based on the target network, the IMCs might have a policy indicating that security protections are required since the network's IMV require IMC authentication, integrity and confidentiality protection when crossing the unprotected network to the IMV. Each IMC consults its policy and could choose to:

   a) Send nothing, if unwilling or unable to participate

   b) Send a hello message indicating its availability to respond to requests

   c) Send a subset of the measurements it is able to collect. Because these messages are transported over an untrustworthy network between the TNCS and the IMV, they might require end-to-end security protection.

   d) Send all of its collected measurements. Because these messages are transported over an untrustworthy network between the TNCS and the IMV, they might require end-to-end security protection.

   e) Send previously received and cached assertion attributes possibly protected by end-to-end security protection.

2) TNCS passes attributes sent by IMCs to the remote IMVs that have expressed an interest in the measurement information received from the TNCC. The IMV exists remotely from the TNCS where the IF-T protections have been terminated so the IF-M messages need protection. The remote IMV may require authentication of the sending IMC to decide whether the IMC's information is considered reliable particularly when multiple IMCs are reporting the same attributes but with different values. Each IMV consults its assessment policy and could choose to:

   a) Send no IF-M messages

   b) Request additional measurement information from the IMC(s) by sending one or more IF-M messages. In response to these IMV requests for measurement information, the IMCs repeat step 1 above. This could lead to additional IMV requests in step 2.

   c) Make an assessment decision based on the measurements provided, return the IMV Evaluation Result to the TNCS and take one or more of the following actions:

i) Send no IF-M messages

ii) Send the component level IMV Action Recommendation to the IMCs. This exposes the IMV Action Recommendation to attack on the network between the TNCS and IMV so security protection may be required.

iii) Send remediation instructions to the IMCs – this exposes the remediation instructions to attack on the network between the TNCS and IMV so security protection may be required. Also the IMV may wish to establish a private, authenticated session with a particular IMC to assure that the proper IMC performs the remediation (when multiple IMC for a component exist on an endpoint). In this case the IMC also benefits by being able to authenticate the IMV in case multiple IMVs are providing remediation instructions.

iv) Send assertion attributes to the IMCs. The assertion attributes describe the level of compliance determined by the IMV. This exposes the assertion attributes to various attacks on the unprotected network between the TNCS and IMV so may require security protection to be employed.

Therefore in order to be able to safely send over the network the IF-M messages, possibly including endpoint measurements, remediation instructions, and even assertion attributes, the IF-M protocol needs to provide additional security protections to safeguard the information as well as if the IMV were local to the TNCS and NAA.

2.4 Requirements

Here are the requirements that IF-M must meet in order to successfully play its role in the TNC architecture.

- Flexibility

  The IF-M protocol MUST support all the functions and use cases described in the TNC architecture as they apply to the communications between the IMC and IMV. The IF-M protocol MUST allow either the IMC or IMV to initiate the assessment or reassessment when operating over a usable IF-TNCCS session. When the IMC initiates the assessment, the IMV MUST allow the IMC to proactively send measurements prior to the IMV sending a measurement request.

  The IF-M protocol MUST be capable of supporting multiple round trip message exchanges during an assessment or reassessment. This allows the IMVs to send multiple requests for measurements potentially based on the results of earlier requests (e.g. based on the endpoint's operating system).

  IF-M attributes MUST be capable of containing a wide variety of types of data values including: binary data, encrypted or compressed data, and textual strings. Any string included in IF-M intended for user display MUST be able to be encoded in the user's preferred language (when known). IF-M MUST be able to carry standard defined attributes and/or vendor-defined attributes.

- Efficient

  The TNC architecture delays network access until the endpoint is determined to not pose a security threat to the network based on its asserted integrity information. To minimize user frustration, the IF-M protocol MUST minimize delays and make IF-M communications as rapid and efficient as possible. Efficiency is also important when you consider that some network endpoints are small and low powered, some networks are low bandwidth and/or high latency, and some IF-T protocols only allow one packet in flight.

- Transport Independence

  IF-M protocol MUST be agnostic of the underlying IF-T transport protocol and thus not change in message format when different IF-T protocols are used. However, IMCs and IMVs may alter their level of verbosity (payload size) in the IF-M message when faced with underlying protocols which are bandwidth constrained.

- Extensible

  IF-M protocol and the attributes contained within it MUST be very extensible allowing for additional protocol capabilities and large numbers of attributes to be defined by future specifications. The attribute name space MUST support large numbers (hundreds) of vendor specific attributes for each vendor without collisions and large numbers (hundreds) of standard defined attributes which can be defined over time. Similarly, the IF-M protocol SHOULD be extensible enough to allow a future set of security protections (e.g. end to end authentication, integrity and confidentiality) to be added offering cryptographic protection for sensitive attributes.

- Scalable

  IF-M protocol MUST be capable of housing a large number (hundreds) of attributes in a single message exchange and allow for use of attributes with large attribute values (tens of kilobytes). This capability might not be practical or even necessary for all deployments (e.g. low bandwidth, high latency, time sensitive environments) but should be possible without alteration of the base protocol. IMCs and IMVs may choose to scale back the number and size of the attributes sent based on other factors (IF-T considerations, privacy filters, etc.).

- Backward Compatibility

  IF-M protocol SHOULD be backwards compatible with the existing IF-IMC and IF-IMV APIs and TNCC and TNCS. This requirement primarily covers the outer IF-M message envelopes which are used by the TNCC and TNCS to route messages and by the IF-IMC and IF-IMV APIs which pass the values to the subscribed components.

2.5 Non-Requirements

Here are certain requirements that IF-M explicitly is not required to meet. This list is not exhaustive (complete).

- End to End Security (IMC to/from IMV)

  IF-M protocol MUST provide the capability to protect its messages end to end between the IMC and IMV. This protection MUST guard against active and passive attackers by offering bi-directional authentication, detection of alteration or replay of the messages, and confidentiality of the message contents as mandated by deployment policy. IF-M security protections enable IMVs existing on a system remote from the termination of the IF-T connection at the NAA to have end to end protected communications with IMCs. This could be particularly important when security sensitive information such as remediation instructions are sent in IF-M messages destined for a single IMC.

- Compression

  IF-M protocol will not automatically compress large messages to improve their suitability for particularly network limitations (e.g. bandwidth, latency). This is the responsibility of the IMC and IMV if it is to be provided.

2.6 Assumptions

Here are the assumptions that the IF-M protocol makes about other components in the TNC architecture.

- Reliable Message Delivery

The TNC Client and TNC Server are assumed to provide reliable delivery for IF-M messages sent between the IMCs and the IMVs. In the event that reliable delivery cannot be provided, the TNC Client or TNC Server is expected to terminate the connection.

2.7 Keywords

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 [KEYWORDS]. This specification does not distinguish blocks of informative comments and normative requirements. Therefore, for the sake of clarity, note that lower case instances of must, should, etc. do not indicate normative requirements.

2.8 IF-M Message Diagram Conventions

This specification defines the syntax of the IF-M message header and the standard set of attributes 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 from left to right for each 32-bit quantity traversing the diagram from top to bottom.   Multi-octet fields representing numeric values must be sent in network (big endian) byte order.

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

3 Design Considerations

This section discusses some of the key design considerations for the IF-M protocol and its relationship to IF-TNCCS.

3.1 Parallel Attribute Namespaces

The IF-M and the equivalent PA-TNC protocol have several fields that contain well known, enumerated values that are key to interoperability. In order to provide interoperability in the standard namespaces (IETF and TNC) while allowing for parallel vendor-defined namespaces for other uses, IF-M includes a namespace identifier called a Vendor ID that is the field immediately prior to the field capable of containing a value from multiple namespaces. For example, two of these namespaces are component types (AKA IF-TNCCS's IF-M Subtypes) and attribute types (that are defined below) each having a corresponding Vendor ID field that allows the recipient to know what namespace (e.g. IETF, TCG) is being used.

It is also important that each of the field's namespaces be readily extensible without constant coordination while also avoiding naming conflicts (two independent new specifications each trying to use the same namespace value in the same field for different purposes). This requirement drove the need for a repository of well known values for each interoperable namespace that specification could request additional values. For example, the IETF's IANA maintains a set of values standardized within the IETF and TCG could have a similar repository allowing each organization to release new specifications (or extensions of this protocol) without value collisions since they each control their own namespace allowing the IETF and TCG to have final say on all value assignments. Note that the TCG and IETF can leverage the other's namespace repository where appropriate.

The separation of IETF, TCG and vendor-defined namespaces is achieved by the inclusion of a Vendor ID qualifier prior to each field supporting multiple namespaces. The value used in the Vendor ID qualifier field is the SMI Private Enterprise Number (PEN) maintained by the IANA that identifies the entity that owns the namespace in use for the next field. Entities wishing to define their own namespace can reserve a PEN value by contacting the IANA at http://pen.iana.org/pen/PenApplication.page.

In order to maximize interoperability and avoid duplication of TCG and IETF standard values, this specification will leverage the IETF PA-TNC 1.0 standard values in the IETF's Vendor ID = 0 namespace when possible. The TCG will also maintain a set of TNC oriented values in the TCG standard (Vendor ID = 0x005597) namespace when appropriate. This approach of specifying the use of the IETF namespace for duplicate values while using the TCG namespace for new TCG oriented values allows implementations based solely on the IETF's PA-TNC specification to interoperate with TNC IF-M implementations while still allowing TCG to have additional capabilities (e.g. for TPM integration).

3.2 IMC and IMV Identifiers

When the IF-TNCCS 2.0 (and PB-TNC) protocols are carrying an IF-M message, the IF-TNCCS protocol includes a header (TNCCS-IF-M-Message) housing several fields important to the processing of a received IF-M message. The IF-M Vendor ID and IF-M Subtype described in the IF-TNCCS specification are used by the TNCC and TNCS to route messages to IMC and IMV that have registered an interest in receiving messages for a particular type of component. Also present in the TNCCS-IF-M-Message header is a pair of fields that identify the IMC and IMV involved in the message exchange. The IMC and IMV Identifier fields are used for performing exclusive delivery of messages and as an indicator for correlation of received attributes. See the IF-TNCCS 2.0 protocol specification for more information on these fields.

Correlation of attributes is necessary when an IMC sends attributes describing multiple different implementations of a single type of component during an assessment, so the recipient IMV(s) need to be able to determine which attributes are describing the same implementation.

For example, a single IMC might report attributes about two installed VPN implementations on the endpoint. Because the individual attributes (except the Product Information attribute) do not include an indication of which VPN product they are describing, the recipient IMV needs something to perform this correlation. Therefore, for this example, the single VPN IMC would need to obtain two IMC Identifiers from the TNC Client and consistently use one with each of the VPN implementations reported during an assessment. The

VPN IMC would group all the attributes associated with a particular VPN implementation into a single IF-M message and send the message using the IMC Identifier it designates as going with the particular implementation. This approach allows the recipient IMV to recognize when attributes in future assessment messages also describe the same VPN implementation.

4 IF-M Message Protocol

This section discusses the use of the IF-M message and its attributes within the TNC architecture and specifies the syntax and semantics for the IF-M message header. The details of each attribute included within the IF-M payload are specified in section 5.

4.1 IF-M Messaging Model

IF-M messages are carried by the IF-TNCCS protocol which provides a multi-roundtrip reliable transport and end to end message delivery to subscribed (interested) parties using a variety of underlying IF-T network protocols. IF-M is unaware of these underlying IF-T transport protocols being used below IF-TNCCS. The interested parties consist of IMCs on the TNCC and IMVs associated with the TNCS that have registered to receive messages about particular types of components (e.g. anti-virus) during an assessment. The IF-M messaging protocol operates synchronously within an assessment session with IMCs and IMVs taking turns sending one or more messages to each other. Each IF-M message may contain zero or more attributes associated with the functional component defined in the IF-TNCCS protocol.

IMCs may only send IF-M messages to IMVs and vice versa. No IMC to IMC or IMV to IMV messaging is allowed to occur. Each IMC or IMV may send several IF-M messages in succession before indicating that it has completed its response to the TNCC or TNCS respectively. As necessary, the TNCC and TNCS will batch these messages prior to sending over the network.

IF-TNCCS provides a message 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 TNCC or TNCS. This is beneficial since it allows one IMC or IMV to combine multiple functions (like anti-virus and personal firewall) by subscribing to both TNC standard component types and also allows multiple IMCs or IMVs to support the same components such as two anti-virus IMVs 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 an IMC or IMV initially sends an IF-M message, it does not know whether it will receive many, one or no IF-M 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 IMC and IMV pair is necessary. For example, an IMV may wish to provide remediation instructions to a particular IMC that it knows is capable of remediating a non-compliant component. This can be accomplished using the IF-TNCCS capability to limit distribution of a message to a single IMC.

4.2 IF-M Relationship to IF-TNCCS

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

One important linkage between the IF-M and IF-TNCCS protocols is the IF-M Subtype that is used by the TNCC and TNCS to route messages to interested IMCs and IMVs. The IF-M Subtype indicates the software component (component type) that is associated with the attributes included inside the IF-M message. Therefore, IMCs and IMVs written to support an assessment of a particular component can register to receive messages about the component and thus participate in its assessment. Each IMC and IMV MUST only send IF-M messages containing attributes that pertain to the software component defined in the message type of the message. This restriction ensures that only the appropriate IMCs and IMVs that support a particular type of component will receive attributes related to that component. If a 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 comprised of two fields: an IF-M Vendor ID and an IF-M Subtype. The Vendor ID identifies the vendor or other organization that defined this message type. The message subtype identifies the message type more particularly within the set of message types defined by that vendor. This specification defines several standard message subtypes to be used with the IETF's Vendor ID to maximize interoperability. Within this specification, the IF-M Subtype field is used to indicate the type of component (e.g. firewall) involved with the message's attributes. Therefore for clarity the IF-M Subtype field will be referred to as the "component type" in this specification. Vendor-defined name spaces may use other semantics for the subtype field as this is outside the scope of this specification.

```
                           1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         IF-TNCCS Header                       |
      |                      Includes Overall Length                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           IF-TNCCS Message of type TNCCS-IF-M-Message         |
      |          Includes IF-M Vendor ID, IF-M Subtype and other      |
      |          fields used by TNCC and TNCS for message routing     |
===============================================================================
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                     IF-M Message Header                       |
      |                Includes Version & Message ID                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         IF-M Attribute                        |
      |                   (e.g. Product Information)                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         IF-M Attribute                        |
      |                   (e.g. Operational Status)                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                       . . . . . .                             |
```

Overview of an IF-TNCCS batch that contains an IF-M Message

For example, if a TNCC sent an IF-TNCCS batch that contained an IF-M message with a component type indicating firewall component, this message would be routed by the TNCS to IMVs registered to assess firewalls. Each registered IMV would receive a copy of the IF-M message including the IF-M header and set of attributes. It is important that each of the attributes included in the IF-M message be associated with the firewall component because only the firewall interested IMC and IMV will receive the message. For example, if the above message contained both firewall and operating system attributes (inside an IF-M message with a component type of firewall), then any IMC and IMV registered to receive operating system messages would not receive those attributes as the messages would only be delivered to those registered for firewall messages.

4.3 IF-M Messages in IF-TNCCS

As depicted in section 4.2, an IF-M message consists of an IF-M header followed by a sequence of zero or more attributes. The IF-M message header (described in section 4.5) and the header for each of the IF-M attributes (specified in section5.1) have a fixed type-length-value (TLV) format. Each IF-M message MAY contain a mixture of standards based and vendor defined attributes identifiable using the type portion of the attribute header. All IMCs and IMVs compliant with this specification MUST be capable of processing multiple attributes in a received IF-M message. IMCs or IMVs that receive an IF-M 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.

4.4 IF-M Component Types

This section defines the component type values used within the IF-TNCCS's IF-M Subtype field for IF-M messages that have the TCG SMI Private Enterprise Number (PEN). These TNC standard values are present in the IF-M Subtype field of the IF-TNCCS protocol (TNCCS-IF-M-Message) to describe which type of component is associated with the IF-M attributes included within the message. This component type field is used by the TNCC and TNCS to route IF-M messages to IMCs and IMVs that have registered to receive messages containing the component type (IF-M Subtype in IF-TNCCS). This allows for IMCs and IMVs to receive messages about specific types of components. More component types will likely be added in the future potentially in different specifications.

Component Name	TNC Standard Component Type	Description
Reserved	0x00000000	Reserved for use in specification examples, experimentation and testing.

TNC Standard Component Types

Note that this table is now nearly empty. The IETF NEA working group has adopted the set of component types included in the original public review version of this specification, so rather than duplicate those values in the TNC namespace the TNC standard will leverage the IETF namespace. The IETF defined component types include: Operating System, Anti-Virus, Anti-Spam, Anti-Malware, Firewall, IDPS, VPN and NEA Client. TNC implementations wishing to assess a component type listed in the prior sentence MUST use the component type values defined in section 3.5 of PA-TNC specification with an IF-M Vendor ID of zero (0 is the IETF namespace). This requirement was included to increase interoperability by forcing implementations of both standards to use the same reserved values. Note that the "NEA Client" component is analogous to the "TNC Client", so IMVs assessing a TNC Client MUST use the NEA Client component type and an IF-M Vendor ID of zero. It is envisioned that future TNC specifications will assign values from the TCG namespace.

4.5 IF-M Message Header Format

This section describes the format and semantics of the IF-M header. Every TNC standard IF-M message MUST start with an IF-M header. The IF-M header provides a common context applying to all of the attributes contained within the IF-M payload. The payload consists of a sequence of assessment attributes described in section 5.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message Identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Version	This field indicates the version of the format for the IF-M message. This version is intended to allow for evolution of the IF-M message header and payload in a manner that can easily be detected by message recipients. IF-M message senders MUST set this field to 0x01 for all IF-M messages that comply with formats and requirements described in version 1.0 of this specification. Implementations responding to an IF-M message containing a

	supported version MUST use the same Version number to minimize the risk of version incompatibility. **Message recipients MUST NOT interpret the contents (after the Version field) of an IF-M message containing a version number that the recipient does not support. . Message recipients MUST respond to an IF-M message containing an unsupported version by sending an IF-M Version Not Supported error in an IF-M Error attribute that is the only IF-M attribute in a IF-M message with version number 1.** **IF-M message initiators supporting multiple IF-M protocol versions SHOULD be able to alter which version of IF-M message they send based on prior message exchanges with a particular peer IMC or IMV.**
Reserved	**Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.**
Message Identifier	**This field contains a value that uniquely identifies the message from a particular IF-M message sender during the assessment. This value can be included in a response message to indicate which message was received and caused the response. For example, this field is included in the TNC error messages so the recipient can determine which message caused the error.** **IF-M message senders MUST NOT send the same message identifier during an assessment. Message identifiers may be randomly generated or sequenced as long as values are not repeated during an assessment message exchange. IF-M message recipients are not required to check for duplicate message identifiers.**

5 IF-M Attributes

This section defines the IF-M attributes that can be carried within an IF-M message. The initial section defines the standard attribute header that appears at the start of each attribute in an IF-M message. The second section defines each of the TNC standard attributes and the final section discusses how vendor-defined attributes can be used within an IF-M message. Vendor-defined attributes are those defined outside of this specification and use the vendor's SMI Private Enterprise Number in the Attribute Type field.

An IF-M message MUST contain an IF-M header (defined in section 4.5) followed by a sequence of zero or more IF-M attributes. All IF-M attributes MUST begin with a standard IF-M attribute header, as defined in section 5.1. The contents of IF-M attributes vary widely, depending on their attribute type. Section 5.2 defines the TCG standard attributes. Section 5.3 discusses how vendor-specific attributes can be defined.

5.1 IF-M Attribute Header

Following the IF-M message header is a sequence of zero or more attributes. All IF-M attributes MUST begin with the standard IF-M 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 IF-M message MUST use the same TLV format. Using the IF-M message version number to indicate the format of all TLV attributes within an IF-M message allows for future versioning of the TLV format in a manner detectable by IF-M message recipients. Similarly, requiring all TLV attribute formats to be the same within an IF-M message also assures that recipients compliant with a particular IF-M message version can at least parse every attribute header and use the length to skip over unrecognized attributes. Finally, all attribute TLVs within an IF-M message MUST pertain to the same implementation of the component. This restriction is relevant when a single IMC is reporting on multiple implementations of a component, so must send multiple IF-M messages each including only the attributes describing a single implementation. For more information on how IMCs should handle multiple implementations see section 3.2.

Every IF-M version 1.0 compliant TLV attribute MUST use the following TLV format:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |          Attribute Type Vendor ID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Attribute Type                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Attribute Length                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Attribute Value (Variable Length)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TLV Field	**Description**

<table>
<tr>
<td>Flags</td>
<td>
<b>This field defines flags impacting the processing of the associated attribute. The following table defines the bit encodings for each flag starting from the left to right:</b>
<table>
<tr><th>Bit Encoding</th><th>Description</th></tr>
<tr>
<td>Bit 0
No Skip Flag
(NOSKIP)</td>
<td><b>Recipients MUST NOT skip this attribute if unknown.
This flag does not mean that all IMCs and IMVs must support this attribute. Instead, any IMC or IMV that receives an attribute with this flag set to one but does not support this attribute MUST NOT process any part of the IF-M message and SHOULD return a IF-M Attribute Type Not Supported error in an IF-M error message.
In order to avoid taking action on a subset of the attributes only to later find an unsupported NOSKIP flagged attribute, recipients of a multi-attribute IF-M message might need to scan all of the attributes prior to acting upon any attribute.
When set to zero, recipients SHOULD skip any unsupported attributes and continue processing the next attribute.</b></td>
</tr>
<tr>
<td><b>Bit 1-7</b></td>
<td><b>Reserved for future use. These bits MUST be set to 0 on transmission and ignored upon reception</b></td>
</tr>
</table>
</td>
</tr>
<tr>
<td>Attribute Type Vendor ID</td>
<td><b>This field indicates the owner of the name space associated with the Attribute Type. This is accomplished by specifying the 24 bit SMI Private Enterprise Number Vendor ID of the party who owns the Attribute Type name space. When an IETF standard attribute is used, this value MUST use the IETF SMI Private Enterprise Number value (0x000000) in this field. For other attributes defined within TCG specifications, this field MUST use the TCG SMI Private Enterprise Number value (0x005597).
The Attribute Type Vendor ID value of 0xffffff is reserved. IMCs and IMVs MUST NOT send IF-M messages in which the Attribute Type Vendor ID has this reserved value (0xffffff). If an IMC or IMV receives a message in which the IF-M Attribute Type Vendor ID has this reserved value (0xffffff), it SHOULD respond with an Invalid Parameter error code in a IF-M Error attribute.</b></td>
</tr>
</table>

Attribute Type	**This field defines the type of the attribute within the scope of the specified vendor name space (Attribute Type Vendor ID) included in the Attribute Value field. The specific TNC standard values allowable in this field when the Vendor ID is the TCG SMI Private Enterprise Number value (0x005597) are defined in section 5.2.** **IMCs and IMVs MUST NOT require support for particular vendor-defined Attribute Types and MUST interoperate with other parties despite any differences in the set of vendor-defined Attribute Types supported (although they MAY permit administrators to configure them to require support for specific IF-M attribute types).** **The Attribute Type value 0xffffffff is reserved. IMCs and IMVs MUST NOT send IF-M messages in which the Attribute Type has this reserved value (0xffffffff). If an IMC or IMV receives a message in which the Attribute Type has this reserved value (0xffffffff), it SHOULD respond with an Invalid Parameter error code in an IF-M Error attribute.**
Attribute Length	**This field contains the length in octets of the entire Attribute including the Attribute's TLV header. Therefore this value MUST always be at least 12. Implementations that do not support the specified Attribute Type can use this length to skip over the attribute to the next attribute. Note that while this field is 4 octets the maximum usable attribute length is less than 2^32 due to limitations of the underlying protocol stack. Specifically, IF-TNCCS's length field includes 32 bytes of other headers which reduce the maximum size available to IF-M since they both use 4 octet length fields.**
Attribute Value	**This field varies depending on the particular type of attribute being expressed. The contents of this field for each of the TNC standard based attributes are defined in section 5.2. For additional information about attributes that are also defined in the IETF, see the IETF PA-TNC specification section 4.2.**

5.2 TNC Standard Attributes

This section discusses the use of the IF-M 1.0 set of TNC standard attributes and their relationship to the IETF namespace. Presently this specification does not define any TNC specific standard attribute types but rather leverages those defined in the IETF's PA-TNC 1.0 specification.

The IETF NEA working group decided to adopt the set of TNC attribute types specified by the public review version of this specification in the equivalent PA-TNC standard. Therefore this specification will not duplicate those attribute definitions in the TCG namespace. These common attribute types include: Attribute Request, Product Information, Numeric Version, String Version, Operational Status, Port Filter, Installed Packages, PA-TNC Error (AKA IF-M Error in TNC context), Assessment Result, Remediation Instructions plus additional attribute types defined in the IETF: Forwarding Enabled, and Factory Default Password. TNC implementations wishing to use the attribute types listed in the prior sentence MUST use corresponding IETF standard attribute values defined in section 4.2 of the PA-TNC specification with an Attribute Type Vendor ID of zero (0 is the IETF namespace).

This requirement to use the IETF namespace is included to increase interoperability by requiring implementations of both standards to use the same reserved values. Note that the “PA-TNC Error” attribute is analogous to the TNC’s IF-M Error, so TNC implementations needing to send an error attribute MUST use the PA-TNC Error attribute with an Attribute Type Vendor ID of zero. It is envisioned that future TNC specifications will define TCG-oriented attribute types in the TCG namespace while continuing to leverage attributes defined in the IETF.

The following subsections discuss the usage, format and semantics of the Attribute Value field for each type of the TNC IF-M 1.0 and IETF PA-TNC 1.0 standard attributes. These fields are included in the variable length Attribute Value field.

5.2.1 Attribute Applicability

This section summarizes which of the attribute types are required to be supported by IMC and IMV supporting each of the types of components described in this specification. IMCs and IMVs associated with a particular type of component SHOULD NOT support additional TNC standard attributes to avoid vagueness in how they are interpreted (e.g. returning a Port Filter attribute for an Anti-Virus component). However IMCs and IMVs are not required to support all the attributes applicable to their component.

The following tables indicates whether an IMC or IMV claiming full support for a particular component type needs to support many of the defined standard attribute types. Every IMC and IMV SHOULD support the Attribute Request attribute allowing an IMV to request particular attributes. Note that the choice of whether an IMC wishes to send each type of attribute to an IMV should be under the control of local privacy policy.

The first table defines the attributes for each particular component type (shown as rows) that an IMC is required to be able to send and an IMV is required to be able to receive during an assessment. The attributes in the IMC attribute support table MUST NOT be sent by IMV.

	Product Info.	**Numeric Version**	**String Version**	**Oper, Status**	**Port Filter**	**Installed Pkgs**	**Forward. Enabled**	**Factory Default Pwd Enabled**	**IF-M Error**
Operating System	MUST	SHOULD	MUST	MAY	MUST NOT	SHOULD **[1]**	SHOULD	SHOULD	**MUST**
Anti-Virus	MUST	MAY	MUST	SHOULD	MUST NOT	SHOULD	SHOULD NOT	SHOULD NOT	**MUST**
Anti-Spyware	MUST	MAY	MUST	SHOULD	MUST NOT	SHOULD	SHOULD NOT	SHOULD NOT	**MUST**
Anti-Malware	MUST	MAY	MUST	SHOULD	MUST NOT	SHOULD	SHOULD NOT	SHOULD NOT	**MUST**
Firewall	MUST	MAY	MUST	SHOULD	SHOULD	SHOULD	SHOULD NOT	SHOULD NOT	**MUST**
Intrusion Detection/ Prevent.	MUST	MAY	MUST	SHOULD	MUST NOT	SHOULD	SHOULD NOT	SHOULD NOT	**MUST**
Virtual Private Network	MUST	MAY	MUST	MAY	SHOULD	SHOULD	SHOULD NOT	SHOULD NOT	**MUST**
NEA (or TNC) Client	**MUST**	**MAY**	**MUST**	**SHOULD**	**MUST NOT**	**SHOULD**	**SHOULD NOT**	**SHOULD NOT**	**MUST**

IMC Sent Attribute Support Requirements for each Component Type

[1] – Support for returning all the installed packages for an entire operating system should be supported by operating system IMC, however deployers should be aware that it is discouraged due to the size of the message and the potential resulting timeouts in underlying transports such as 802.1X. IF-T transport such as IF-T Binding to TLS would be able to send all the package information, so local IMC policy should dictate the circumstances when the operating system IMC should send the package information.

The following table defines the attributes for each particular component type (shown as rows) that an IMC is required to be able to receive and an IMV is required to be able to send during an assessment. The attributes in the following table MUST NOT be sent by IMC.

	IMV Assess. Results	**Remed. Instruct**	**IF-M Error**
Operating System	SHOULD	MAY	**MUST**
Anti-Virus	SHOULD	MAY	**MUST**
Anti-Spyware	SHOULD	MAY	**MUST**
Anti-Malware	SHOULD	MAY	**MUST**
Firewall	SHOULD	MAY	**MUST**
Intrusion Detection/ Prevent.	SHOULD	MAY	**MUST**
Virtual Private Network	SHOULD	MAY	**MUST**
NEA (or TNC) Client	**SHOULD**	**MAY**	**MUST**

IMV Sent Attribute Support Requirements for each Component Type

5.2.2 Attribute Request

The Attribute Request allows an IMV to request certain attributes from the registered set of IMCs. All IMC that implement any of the TCG standard component types defined in this specification SHOULD support receiving and processing this attribute type for at least the TCG standard IF-M Subtypes. Similarly, all IMVs that implement any of the TCG standard components defined in this specification SHOULD support sending this attribute type at least for those IF-M Subtypes. IMVs MUST NOT include this attribute type in an Attribute Request attribute. It does not make sense for a IMV to request that an IMC send an Attribute Request attribute.

The registered IMCs MAY choose to send all, a subset or none of the request attributes but MUST NOT send attributes that were not requested (except error attributes). Each Attribute Request MUST contain at least one vendor-defined or TNC standard attribute type. Because the length of a Vendor ID paired with an Attribute Type has a fixed length of 8 octets, the number of requested attributes can be computed using the Attribute Length field (in the Attribute Header).

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 example 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 IF-T transport protocol).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reserved    |           Attribute Type Vendor ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Attribute Type                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reserved    |           Attribute Type Vendor ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Attribute Type                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	**Description**
Reserved	**Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.**
Attribute Type Vendor ID	**This field contains the SMI Private Enterprise Number of the vendor who controls the name space for the following Attribute Type. This field enables vendor and standards based attributes to be used without potential collisions.** **The TNC standard attributes defined in section 5.2 MUST use the TCG SMI Private Enterprise Number (0x005597) in this field unless inherited from the PA-TNC standard. Vendor-defined attributes MUST use the SMI Private Enterprise Number of the vendor who defined the attribute. IETF standard attributes MUST use the IETF SMI Private Enterprise Number (0x000000) in this field.**
Attribute Type	**The Attribute Type field (together with the Attribute Type Vendor ID) indicates the specific attribute requested. The TNC standard Attribute Types defined in section 5.2 that have a security or measurement data related purpose can be requested using this field. 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 IMC or IMV that receives an Attribute Request containing one of the prohibited Attribute Types SHOULD respond with an Invalid Parameter error in an IF-M error message.**

5.2.3 Product Information

This attribute contains vendor and product level information about the product that implements the component specified in the component type field in the IF-TNCCS header (see section 3.2) of a TNC standard IF-M message. For example, if the component type is anti-virus, this attribute would contain information about the anti-virus product installed on the endpoint.

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.

```
                     1                   2                   3
```

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Product Vendor ID               |  Product ID   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Product ID   |         Product Name (Variable Length)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           . . . . . . .                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Product Vendor ID	This field contains the IANA assigned SMI Private Enterprise Number for the organization that created the product. If the product creator does not have an SMI Private Enterprise Number or is unknown, this value MUST be set to zero (0) and the identity of the product creator SHOULD be included in the Product Name along with the name of the product. The current list of SMI Private Enterprise Number assignments can be found at: http://www.iana.org/assignments/enterprise-numbers.
Product ID	This enumeration uniquely identifies the product containing the requested component from the product's vendor. If the product vendor is unknown, the Product ID field MUST be 0. This field is a vendor-defined field to identify the particular component present on the endpoint. For example, Symantec offers numerous anti-virus oriented products. If the request was for an anti-virus component, this enumeration could be used to identify which anti-virus product is present on the system. 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 string identifying the product (e.g. "Symantec Norton AntiVirus™ 2008") in enough detail to unambiguously distinguish it from other products from the vendor. This might require inclusion of information about the edition or other product marketing information to assure it is unambiguously identified. Products associated with a known vendor who does not have a registered SMI Private Enterprise Number SHOULD be represented using a combination of the vendor name and full product name (e.g. "Ubuntu® IPtables" for the IPtables firewall in the Ubuntu distribution of Linux). The length of this field can be determined by starting with the Attribute Length field in the attribute header and subtracting the size of the fixed length fields (12 for the Attribute Header and 5 for this Attribute's fields) that precede it. However, implementers should be careful that the Attribute Length is not less than the size of the fixed length fields. Such a circumstance could cause a buffer overflow if not handled properly. It is a syntax error and should result

	in a TNC Invalid Parameter error code.

5.2.4 Numeric Version

This attribute describes the detailed version information about the requested component (e.g. operating system) in use on the endpoint. This version includes structured values for the version information to enable IMVs to perform comparative operations on the version. The version information included is associated with a particular product, so IMV are expected to also possess the corresponding Product Information attribute when interpreting this attribute. Some IMC may not be able to determine some or all of this information for its component. Similarly many components do not use such granular version information. It's envisioned that this attribute could be useful for describing the version of the operating system.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Major Version Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Minor Version Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Build Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Service Pack Major       |      Service Pack Minor       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	**Description**
Major Version Number	**This field contains the major version number for the component (e.g. Windows® Vista is 6). For operating systems, this value can be obtained using APIs like GetVersionEx on Windows and uname on Solaris. If unused or unknown, this field SHOULD be set to zero (0).**
Minor Version Number	**This field contains the minor version number for the component (e.g. Solaris™ 10 is 10). For operating systems, this value can be obtained using APIs like uname on Linux and oslevel on AIX. If unused or unknown, this field SHOULD be set to 0.**
Build Number	**This field contains the internal engineering group's build number. This provides more granularity than the minor version number as many builds might occur leading up to an official release major/minor version. For operating systems, this value can be obtained using APIs like GetVersionEX on Windows and uname on Linux. If this field is not used or unknown, the value SHOULD be set to zero (0).**
Service Pack Major	**If applicable, this field contains the service pack major version number as provided by an API like GetVersionEX on Windows. If this field is not used or unknown, the value SHOULD be set to zero (0).**

Service Pack Minor	**If applicable, this field corresponds to the Service Pack Major above but provides more granularity into the service pack version. If this field is not used or unknown, the value SHOULD be set to zero (0).**

5.2.5 String Version

This attribute contains the version information of the component defined in the component type field in the IF-TNCCS header (see section 4.2) for a TNC standard IF-M message. The version information included is associated with a particular product, so IMVs are expected to also possess the corresponding Product Information attribute when interpreting this 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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version Len  |  Component Version Number (Variable Length)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Build Num Len |   Internal Build Number (Variable Length)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Config. Len  | Configuration Version Number (Variable Length)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Version Len	**This field defines the number of octets in the Component Version Number string 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).**
Component 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 for version strings. Some use alphabetic characters, white space, and punctuation. Therefore, the syntax and semantics of this version string are not defined.**
Build Num Len	**This field defines the number of octets in the Internal Build Number string field. For components where the internal build number is unavailable or unknown, this field MUST be set to zero and the Internal Build Number is not present.**
Internal Build Number	**This field contains a UTF-8 string representing the vendor internal engineering build number of the product. In some cases this value is used to differentiate different minor (or test) releases of a product prior to declaring a new official version release. The syntax and semantics of this string are not defined.**

Config. Len	**This field defines the number of octets in the Configuration Version Number string 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).**
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. IMCs MAY include multiple versions 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. Therefore, the syntax and semantics of this version string are not defined.**

5.2.6 Operational Status

This attribute describes the operational status of the component defined in the component type field in the IF-TNCCS header (see section 4.2). For example, if the IF-M Subtype (component type) 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.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Status     |     Result    |         Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Last Use                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Last Use (continued)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Last Use (continued)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Last Use (continued)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Last Use (continued)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	**Description**

<table>
<tr>
<td>Status</td>
<td><strong>Operational status of the usability of the component.</strong>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td><strong>Status is unknown or other</strong></td></tr>
<tr><td>1</td><td><strong>Not installed on system</strong></td></tr>
<tr><td>2</td><td><strong>Installed but not operational</strong></td></tr>
<tr><td>3</td><td><strong>Operational</strong></td></tr>
<tr><td><strong>4+</strong></td><td><strong>Reserved for future use</strong></td></tr>
</table>
<strong>If an IMV receives a value for this field that it does not recognize, it SHOULD treat this value as equivalent to the value 0.</strong></td>
</tr>
<tr>
<td>Result</td>
<td><strong>This field contains the result of the last use of the component. IMCs MUST set this field to zero when the Status field contains a value of 1 (Not installed) or 2 (Installed but not operational). The following table enumerates the values of this field:</strong>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td><strong>Status is unknown or other</strong></td></tr>
<tr><td>1</td><td><strong>Successful use with no errors detected</strong></td></tr>
<tr><td>2</td><td><strong>Successful use with an error detected</strong></td></tr>
<tr><td>3</td><td><strong>Last use aborted or otherwise unsuccessful</strong></td></tr>
<tr><td><strong>4+</strong></td><td><strong>Reserved for future use</strong></td></tr>
</table>
<strong>If an IMV receives a value for this field that it does not recognize, it SHOULD treat this value as equivalent to the value 0.</strong></td>
</tr>
<tr>
<td>Reserved</td>
<td><strong>Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.</strong></td>
</tr>
<tr>
<td><strong>Last Use</strong></td>
<td><strong>This field contains the date and time of the last use of the component, if known. The Last Use date and time MUST be represented as an RFC 3339[RFC3339] 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 IMVs MUST support them.</strong>
<strong>The Last Use string MUST NOT be NUL terminated or padded in any way. If the last use is not known, not applicable, or cannot be represented in this format, this field MUST contain ”0000-00-00T00:00:00Z” allowing this attribute to be fixed length. Note that this reserved value is not RFC 3339 compliant (zero month).</strong>
<strong>This encoding produces an easy to read, parse and interpret string in YYYY-MM-DDTHH:MM:SSZ format that can precisely define a particular second in UTC time. For example, 9:05:00AM EST on January 19, 1995 can be represented as</strong></td>
</tr>
</table>

	“1995-01-19T14:05:00Z”. The length of this field is always 20 octets.

5.2.7 Port Filter

This attribute includes the list of port numbers and their associated protocols (e.g. TCP and UDP) that are currently blocked or allowed by the host-based firewall on the endpoint. Each Protocol and Port Number combination uses 4 octets, so the number of filtered ports can be calculated using the Attribute Length in the Attribute Header (see section 4.1).

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 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 IF-T transport 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 IF-M Attribute Length field by subtracting the number of octets in the IF-M Attribute Header and dividing by 4. If the Attribute Length field is invalid, IMVs SHOULD respond with an Invalid Parameter IF-M error code.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reserved  |B|    Protocol   |         Port Number           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reserved  |B|    Protocol   |         Port Number           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	**Description**
Reserved	**Reserved for future use. This field MUST be set to zero 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 one if the protocol/port combination is blocked otherwise this field MUST be set to zero. 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, this could just list the few as not blocked).** **IMCs MUST NOT provide a mixed list of block and non-blocked ports for a particular protocol. IMCs MUST NOT list the same Protocol and Port Number combination twice in an attribute. IMCs MAY list all blocked ports for one protocol and all allowed ports for a different protocol in this attribute using the B flag to indicate whether each are blocked.** **For example, an IMC 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 mirror those of the IPv4 Protocol and IPv6 Next Header fields. The allowable values for this field are managed by the IANA. The current list of transport protocol values can be found at: http://www.iana.org/assignments/protocol-numbers**
Port Number	**This field contains the transport protocol (e.g. tcp) port number being blocked or allowed. This field mirrors the port allocation space assigned for the above specified Protocol field. For example, if the Protocol is TCP (6) then the port numbers are those associated with TCP. The TCP and UDP port number assignments are managed by the IANA and can be found at: http://www.iana.org/assignments/port-numbers**

5.2.8 Installed Packages

This attribute contains meta-data about installed packages that comprise a particular component. This allows an IMV to check the versions of packages that are installed for a particular component and which versions of those packages are installed.

This attribute type can be quite long, especially for the Operating System component type. This length can cause problems, especially with 802.1X and other limited transport protocols. Therefore, IMCs SHOULD NOT send this attribute unless specifically requested to do so using the Attribute Request attribute or otherwise configured to do so. Also, IMVs SHOULD NOT request this attribute unless the transport protocol in use can support the large amount of data that may be sent (e.g. IF-T Binding to TLS) in response.

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 IF-T transport protocol can support that many). If this number is not sufficient, specialized patch management software should be employed which can simply report compliance with a pre-established patch policy.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |         Package Count         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Pkg Name Len  |        Package Name (Variable Length)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version Len  |    Package Version Number (Variable Length)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	**Description**
Reserved	**Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.**
Package Count	**This field indicates the number of packages described in this attribute. Each package description includes both the**

	variable length package name (Pkg Name Len, Package Name) and its version (Version Len, Package Version Number) as these are always both present for each described package.
Pkg Name Len	**This field indicates the length of the Package Name in octets. This field may be zero if a Package Name is not available.**
Package Name	**This field contains a UTF-8 string identifying the name of the package associated with the type of component. This field MUST be sized to fit the version string and MUST NOT include extra octets for padding or NUL character termination. This field's content is package technology dependent (e.g. RPM names on Linux) therefore the syntax and semantics of this name are not specified in this document, since they may vary across products and/or operating systems. IMCs 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 indicates the length of the Package Version Number in octets. This field may be zero if a Package Version Number is not available.**
Package Version Number	**This field contains a UTF-8 string identifying the version (e.g. "1.2.3.4") of the package named by the prior field in this attribute. This field MUST be sized to fit the version string and 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. IMCs 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.**

5.2.9 IMV Assessment Result

This attribute contains the final assessment result and action recommendation from a particular IMV. This value might be returned to an IMC for information purposes (e.g. when an assessment is successful) or in conjunction with other attributes indicating that corrective action is required. 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 IF-M attributes such as Remediation Instructions sent to the IMC.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      IMV Assessment Result                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

<table>
<tr><th>Header Field</th><th>Description</th></tr>
<tr><td>IMV Assessment Result</td><td>This field contains the numeric evaluation result returned by the IMV to the TNCS. This field MUST take one of the TNC standard values from the following table:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>IMV’s component was assessed to be compliant with policy (TNC_IFM_EVALUATION_RESULT_COMPLIANT)</td></tr>
<tr><td>1</td><td>IMV’s component was assessed to be non-compliant with policy but the difference from compliance was minor. (TNC_IFM_EVALUATION_RESULT_NON_COMPLIANT_MINOR)</td></tr>
<tr><td>2</td><td>IMV’s component was assessed to be non-compliant and the assessed difference was very significant. (TNC_IFM_EVALUATION_RESULT_NON_COMPLIANT_MAJOR)</td></tr>
<tr><td>3</td><td>IMV was unable to determine policy compliance due to an error (TNC_IFM_EVALUATION_RESULT_ERROR)</td></tr>
<tr><td>4</td><td>IMV was unable to determine whether IMC measurements are compliant with policy. (TNC_IFM_EVALUATION_RESULT_DONT_KNOW)</td></tr>
</table>
</td></tr>
</table>

5.2.10 Remediation Instructions

The Remediation Instructions attribute is sent by the IMV to the IMC(s) and contains remediation instructions for updating a particular component to make the endpoint compliant with the assessment policies. An IMV might choose to send more then one Remediation Instructions attributes in some circumstances (e.g. both a URI and a human readable message) to remediate one or more components. Because many remediation approaches exist, one goal for this attribute was to create a single extensible attribute type capable of carrying instructions for a wide variety of standards-based and vendor-defined remediation technique. This single attribute type approach also eases the TNCC or TNCS’s ability to recognize and potentially block when an IMV is requesting an IMC remediate a particular component in case this wasn’t desired.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Reserved   |       Remediation Parameters Vendor ID        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Remediation Parameters Type                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Remediation Parameters (Variable Length)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Reserved	Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.

<table>
<tr><td>Remediation Parameters Vendor ID</td><td>This field contains the IANA assigned SMI Private Enterprise Number for the vendor whose Remediation Type name space is being used in the attribute. For TCG standards based Remediation Parameters Type values this field MUST be set to 0x005597. For IETF standards based Remediation Parameters Type values this field MUST be set to 0x000000. For other vendor-defined types of remediation, this field MUST contain the vendor’s SMI Private Enterprise Number.</td></tr>
<tr><td>Remediation Parameters Type</td><td>This field identifies the format and semantics of remediation parameters (instructions) contained within the attribute. This type exists within the scope of Remediation Parameters Vendor ID defined name space allowing for both vendor-defined and TCG standard name spaces. In order to increase interoperability, this specification leverages the IETF defined remediation parameter types defined in section 4.2.10 of the PA-TNC standard.

When the Remediation Parameters Vendor ID is set to the IETF Private Enterprise Number (0), the following table lists the supported Remediation Type values:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Invalid value (MUST NOT be used)</td></tr>
<tr><td>1</td><td>TNC URI-Based Remediation</td></tr>
<tr><td>2</td><td>UTF-8 Encoded String</td></tr>
<tr><td>3+</td><td>Reserved for future use</td></tr>
</table>
</td></tr>
<tr><td><b>Remediation Parameters</b></td><td>This field varies depending on the particular type of remediation parameters being expressed. The contents of this field for each of the TCG (and IETF) standard based remediation instructions are defined in the following subsections.</td></tr>
</table>

5.2.10.1 TNC Remediation Parameters

The following subsection specifies the TCG standard format that MUST be used in the Remediation Parameters field when the Remediation Vendor ID is set to the IETF SMI Private Enterprise Number of zero. Additional TNC standard remediation instruction types are envisioned to be added in future revisions of this specification that might leverage the TCG SMI Private Enterprise Number.

5.2.10.2 TNC URI-Based Remediation

This attribute provides information to facilitate a TNC standard, semi-manual remediation where a human could be required to take a corrective action using the provided URI to the remediation server.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Remediation URI (Variable Length)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Remediation URI	**This field contains a URI referencing the service capable of providing the remediation updates to the system. 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. This URI MUST be converted to a UTF-8 sequence of octets and then percent encoded where necessary [RFC3986]. IMCs should validate that the URI and instructions come from a trustworthy source to avoid being tricked into performing damaging actions.**

5.2.10.3 TNC UTF-8 String Remediation

This attribute provides information to facilitate a TNC standard, semi-manual remediation where a message needs to be displayed to a human to take a corrective action.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Remediation String Length                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Remediation String (Variable Length)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Lang Code Len |  Remediation String Lang Code (Variable Len)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Remediation String Length	**This field contains the length of the Remediation String in octets.**
Remediation String	**This field MUST contain a UTF-8 encoded string containing information for possible display to the user in order to enable a remediation of the endpoint component. This field MUST be sized to fit the remediation string and MUST NOT include extra octets for padding or NUL character termination. This string should contain human-readable instructions for remediation that MAY be displayed to the user by the IMC. This attribute may be useful in conjunction with the remediation URI to aid the user to know how to remediate with the provided URI.**
Lang(uage) Code Len(gth)	**This field contains the length in octets of the Remediation String Lang Code field.**
Remediation String Lang(uage) Code	**The Remediation String Lang(uage) Code field contains a US-ASCII string comprised 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.**

5.2.11 Forwarding Enabled

This 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,

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Forwarding Enabled                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

<table>
<tr><th>Header Field</th><th>Description</th></tr>
<tr><td><b>Forwarding Enabled</b></td><td><b>This field indicates whether network traffic is being allowed to be forwarded between network interfaces on the endpoint. Such forwarding could present a security risk.</b>
<b>The following table lists the supported Forwarding Enabled values:</b>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td><b>Endpoint is not forwarding traffic</b></td></tr>
<tr><td>1</td><td><b>Endpoint is allowing traffic to be forwarded</b></td></tr>
<tr><td>2</td><td><b>Unable to determine whether endpoint is forwarding traffic</b></td></tr>
</table></td></tr>
</table>

5.2.12 Factory Default Password Enabled

This attribute indicates whether the endpoint has a factory default password enabled for use. Some types of endpoints include a default static password 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. This attribute is currently envisioned to be specific to reporting posture for the operating system component,

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Factory Default Password Enabled              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description

<table>
<tr><td><b>Factory Default Password Enabled</b></td><td><b>This field indicates whether the endpoint includes a factory default password which is currently enabled for use. Allowing for the use of a default password might present a security risk to the endpoint.</b>

<b>The following table lists the supported Factory Default Password Enabled values:</b>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td><b>Endpoint does not have a factory default password enabled.</b></td></tr>
<tr><td>1</td><td><b>Endpoint does have a factory default password enabled.</b></td></tr>
</table>
</td></tr>
</table>

5.2.13 IF-M Error

This attribute contains error codes and supplemental information regarding IF-M level messaging errors.

An IF-M error SHOULD be sent with the same IF-M Vendor ID and IF-M Subtype used by the IF-M message that caused the error so that the error code is sent to the party who sent the offending IF-M message. Other measures (such as setting IF-TNCCS's EXCL flag and the IMC Identifier or IMV Identifier fields) SHOULD also be taken to attempt to ensure that only the party who sent the offending message receives the error.

When an IF-M error is received, the recipient MUST NOT respond with an IF-M error 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).

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Reserved   |               Error Code Vendor ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Error Code                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Error Information (Variable Length)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Reserved	**Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.**
Error Code Vendor ID	**This field contains the IANA assigned SMI Private Enterprise Number for the vendor whose Error Code name space is being used in the attribute. For TCG standard Error Code values this field MUST be set to 0x005597. For IETF standard Error Code values this field MUST be set to 0x000000. For other vendor-defined Error Code name spaces this field MUST be set to the SMI Private Enterprise Number of the vendor.**

<table>
<tr><td></td><td><strong>In order to maximize interoperability with implementations of the IETF’s PA-TNC, the TNC standard shares the use of IETF’s set of error codes specified in section 4.2.8 of the PA-TNC specification. These codes are listed below for completeness but should be used with an IETF PEN value of 0x000000 in this field. Future TNC specific errors may be defined in future specifications and use the TCG Private Enterprise Number.</strong></td></tr>
<tr><td>Error Code</td><td><strong>This field contains the error code being reported in the attribute. This code exists within the scope of the Error Code Vendor ID defined name space allowing for both vendor-defined and TCG and IETF standard name spaces. IMCs and IMVs MUST NOT require support for particular vendor-specific IF-M Error Codes and MUST interoperate with other parties despite any differences in the set of vendor-specific IF-M Error Codes supported (although they MAY permit administrators to configure them to require support for specific IF-M error codes).</strong>

<strong>When the Error Code Vendor ID is set to the IETF Private Enterprise Number, the following table lists the supported TCG standard numeric error codes:</strong>

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td><strong>Reserved value</strong></td></tr>
<tr><td>1</td><td><strong>IF-M attribute parameter is invalid, unknown or unsupported (TNC_IFM_INVALID_PARAMETER)</strong></td></tr>
<tr><td>2</td><td><strong>IF-M protocol version not supported (TNC_IFM_VERSION_NOT_SUPPORTED)</strong></td></tr>
<tr><td><strong>3</strong></td><td><strong>IF-M attribute unknown or not supported (TNC_IFM_ATTRIBUTE_NOT_SUPPORTED).</strong></td></tr>
</table>
</td></tr>
<tr><td><strong>Error Information</strong></td><td><strong>This variable length value provides additional context for the error. The length of this field can be determined by the recipient using the IF-M header length field.</strong>

<strong>Subsections under 5.2.13.1 show the supplemental error information that MUST be included for each TCG standard error code. This information frequently involves sending a portion of the original IF-M message so the recipient can determine which message caused the error and the messages content.</strong></td></tr>
</table>

5.2.13.1 IF-M Error Structures

The following subsections show the supplemental error information that MUST be included in the Error Information field for each TCG standard error code.

5.2.13.2 IF-M Invalid Parameter Error Information

The TNC Invalid Parameter error code indicates that the sender of this error code has detected an invalid or unknown value in an IF-M message sent by the recipient of this error code in the current assessment.

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.

```
                    1                   2                   3
```

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Message Identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Offset                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	**Description**
Version	**This field MUST contain an exact copy of the Version field in the IF-M Message Header of the IF-M message that caused this error.**
Reserved	**This field MUST contain an exact copy of the Reserved field in the IF-M Message Header of the IF-M message that caused this error.**
Message Identifier	**This field MUST contain an exact copy of the Message Identifier field in the IF-M Message Header of the IF-M message that caused this error.**
Offset	**This field MUST contain an octet offset from the start of the IF-M Message Header of the IF-M message that caused this error to the start of the value that caused this error. For instance, if the first IF-M attribute in the message had an invalid IF-M Attribute Length (e.g. 0), this value would be 16.**

5.2.13.3 IF-M Version Not Supported Error Information

The Version Not Supported error code is a TCG Standard IF-M Error Code that indicates that the sender of this error code does not support the IF-M version number included in the Message Header of an IF-M 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 IF-M message that contained the unsupported version as well as Max Version and Min Version fields that indicate which IF-M version numbers are supported by the sender of the error code.

The sender MUST support all IF-M versions between the Min Version and the Max Version, inclusive. When possible, recipients of this error code SHOULD send future messages to the IMC or IMV that originated this error message with an IF-M version number within the stated range.

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

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |              Copy of Reserved                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Message Identifier                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Max Version  |  Min Version  |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Version	**This field MUST contain an exact copy of the Version field in the IF-M Message Header of the IF-M message that caused this error.**
Copy of Reserved	**This field MUST contain an exact copy of the Reserved field in the IF-M Message Header of the IF-M message that caused this error.**
Message Identifier	**This field MUST contain an exact copy of the Message Identifier field in the IF-M Message Header of the IF-M message that caused this error.**
Max Version	**This field MUST contain the maximum IF-M version supported by the sender of this error code.**
Min Version	**This field MUST contain the minimum IF-M version supported by the sender of this error code.**
Reserved	**Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.**

5.2.13.4 IF-M Attribute Type Not Supported Information

The Attribute Type Not Supported error code is a TCG standard IF-M Error Code that indicates that the sender of this error code does not support the IF-M Attribute Type included in the Error Information field. This unsupported Attribute Type was included in an IF-M message sent by the recipient of this error code in the current assessment. This field MUST contain the initial 8 octets of the IF-M Message Header followed by the initial 8 octets of the IF-M attribute header for the attribute that was not supported.

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.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |            Copy of Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Message Identifier                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |            Attribute Type Vendor ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Attribute Type                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description

Version	**This field MUST contain an exact copy of the Version field in the IF-M Message Header of the IF-M message that caused this error.**
Copy of Reserved	**This field MUST contain an exact copy of the Reserved field in the IF-M Message Header of the IF-M message that caused this error.**
Message Identifier	**This field MUST contain an exact copy of the Message Identifier field in the IF-M Message Header of the IF-M message that caused this error.**
Flags	**This field MUST contain an exact copy of the Flags field in the IF-M Attribute Header of the IF-M attribute that caused this error.**
Attribute Type Vendor ID	**This field MUST contain an exact copy of the Attribute Type Vendor ID field in the IF-M Attribute Header of the IF-M attribute that caused this error.**
Attribute Type	**This field MUST contain an exact copy of the IF-M Attribute Type field in the IF-M Attribute Header of the IF-M attribute that caused this error.**

5.3 Vendor-Defined Attributes

This section discusses the use of vendor-defined attributes within IF-M. The IF-M 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. 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 IF-M without the potential for name space collisions or misinterpretations. In order to avoid collisions, IF-M uses the well-established SMI Private Enterprise Numbers as Vendor IDs to define separate name spaces for important fields within the message. For example, to ensure the uniqueness of message types while providing for vendor extensions, vendor-defined message types include the vendor's unique Vendor ID to indicate the intended name space for the message subtype followed by the message subtype. Message types and attribute types standardized by the TCG will use the TCG or IETF's (for shared values included in the IETF specifications) SMI Private Enterprise Number in the Vendor ID.

SMI Private Enterprise Numbers are used to provide a separate identifier space for each vendor. IANA provides a registry for SMI Private Enterprise Numbers at http://www.iana.org/assignments/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".

6 Security Considerations

This section discusses the major types of potential security threats relevant to the IF-M message protocol and summarizes the expected security protections that should be offered by IF-M security protocol(s). IF-M security protocol(s) are described in separate specifications which layer upon the base IF-M protocol described in this specification. It is envisioned that additional attribute types will be defined to facilitate the exchange of security capabilities, keys, and security protected attributes. Ultimately, the TNC deployer decides whether each particular security protection is necessary for a particular deployment environment, so the expected security protections discussed in this section highlight the need for IF-M security protocol implementations to be capable of offering the feature.

6.1 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 IF-M protocol. Some deployments may wish to reduce the amount of assumed trust by using an IF-M security protocol to protect the IF-M messages. The following sub-sections discuss the trust properties associated with each portion of the TNC architecture directly involved with the processing of the IF-M protocol.

6.1.1 IMC

The IMCs are trusted by IMVs to:

- Collect valid information about the component type associated with the IMC
- Report upon collected information consistent with local security and privacy policies
- Accurately report information associated with the type of component for the IF-M message
- Not act maliciously including not launching denial of service attacks against the IMVs
- Perform specified remediation instructions only when appropriate for IMC’s specific product

6.1.2 IMV

The IMVs are trusted by IMCs to:

- Only request information necessary to assess the security state of the endpoint
- Make assessment decisions based on deployer defined policies
- Return the correct IMV Action Recommendation to the TNCS and when necessary the IMCs
- Discard collected information consistent with its data retention and privacy policies
- Provide accurate Remediation Instructions to involved IMCs when required
- Not act maliciously to TNCS and IMCs including not launching denial of service attacks against their operation
- Not to send malicious remediation instructions that does not fix or cause damage to the endpoint.

6.1.3 TNCC, TNCS and IF-TNCCS

The TNCC and TNCS are trusted by the IMC and IMV to:

- Provide a reliable transport for IF-M messages
- Deliver messages for a particular component type only to those IMCs and IMVs that have registered for them
- Not disclose any provided attributes to parties outside of the TNC assessment

- Not act maliciously to drop, duplicate or flood the IMCs and IMVs with unnecessary messages
- Not to observe, fabricate or alter the contents of an IF-M message (this trust could be minimized with an IF-M security protocol)
- Properly place IMC and IMV identifiers into the IF-TNCCS protocol, deliver those identifiers to IMCs and IMVs as needed, and manage exclusive delivery to a particular IMC or IMV
- Properly expose the identity of the peer TNCC or TNCS for use by IMC and IMV to make policy decisions

6.2 Security Threats

Beyond the trusted relationships assumed in section 6.1, the IF-M protocol faces a number of potential security attacks that could require targeted security countermeasures. IF-M security protocol specification(s) MUST state if and how the security protocol will safeguard against these types of attack.

Generally the IF-M protocol relies upon the underlying IF-T protocol to protect the messages from attack when traveling over the network. Once the message resides on the TNCC or TNCS, it is trusted to be properly and safely delivered to the appropriate IMCs and IMVs.

6.2.1 Attribute Theft

When IF-M messages are sent over unprotected network links or spanning less trusted local software stacks, the contents of the IF-M 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). IF-M 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 other financial gain. Therefore it is important that IF-T provide strong confidentiality protection.

6.2.2 Message Fabrication

Attackers on the network or present within the TNC architecture stack could introduce fabricated IF-M messages intending to trick or cause a denial of service for 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 IMC automatically following the instructions. IMC need to ensure that any requests to take actions on the endpoint (such as remediation instructions) received from IMV(s) are authentic and trustworthy using strong authentication and integrity protections offered by IF-T. IMCs should not blindly follow remediation instructions received from a trusted TNC Server. At least for patches and other potentially dangerous actions, IMCs should validate these actions (e.g. via user confirmation) before proceeding.

Such an attack could occur if an active attacker could launch a man-in-the-middle (MiTM) attack by proxying the IF-M messages and was able to replace undesired messages with ones easing future attack upon the endpoint. For example if IF-T security protection is not used and the TNCS proxies all assessment traffic to a remote TNCS, the proxy could eavesdrop and replace the IMV assessment results attribute tricking the endpoint into thinking it has passed an assessment when in fact it has not and requires remediation. Because the IMC has no way to verify that the assessment results were actually created by an authentic IMV it is unable to detect the falsified attribute or message. Therefore, it is important that IF-T provides strong authentication and integrity protection.

6.2.3 Attribute Modification

This attack could allow an active attacker capable of intercepting a message to modify an IF-M 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 stealthy modify the attribute exchange. For example, an attacker might wish to change the contents of 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 IMC to the current version number so the IMV's assessment passes while leaving the endpoint vulnerable to attack. Similarly an attacker could achieve wide spread denial of service by altering large number of assessments' version string attribute to an old value so they repeatedly fail assessments even after a successful remediation. By sending a lower value the IMV continues to believe that the endpoint is running old, potentially vulnerable versions of the firewall that does not meet network compliance policy so therefore is not allowed to join the network. Use of an IF-T protocol providing strong integrity protection and authentication is essential as countermeasures to these attacks.

6.2.4 Attribute Replay

Another potential attack against an unprotected IF-M 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 measurement 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 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 an IF-T protocol providing strong integrity protection and authentication is essential as countermeasures to these attacks.

6.2.5 Attribute Insertion

Similar to the attribute modification attacks, an adversary wishing to include one or more attributes or IF-M messages inside a valid assessment may be able to insert the attributes or messages without detection by the recipient. Even if authentication of the parties is present during an IF-M exchange, if no per-message and per-session integrity protection is present an attacker can add information to the assessment possibly causing incorrect assessment results. For example an attacker could add attributes to the front of an IF-M 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 an IMC or IMV 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 the IF-T protocol providing strong authentication and integrity protection could prevent the adversary from inserting attributes into the assessment. Use of the IF-T protocol providing strong authentication and integrity protection could prefer the adversary from inserting attributes into the assessment.

6.2.6 Denial of Service

A variety of types of denial of service attacks are possible against the IF-M message exchange if left unprotected to untrusted parties along the communication path between the IMC and IMV. Normally the IF-T exchange is bi-directionally authenticated which helps to prevent MiTM on the network from active proxies 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 TNCC or TNCS that appear to be part of an assessment these messages could confuse or cause the IMC and IMV to perform unnecessary work or take incorrect action. Several example denial of service situations are described in section 6.2.3 and 6.2.5. Many potential denial of service examples exist including flooding messages to IMC or IMV, sending very large messages containing many attributes, and repeatedly asking for resource intensive operations.

7 Privacy Considerations

The IF-M 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 IMVs and TNCS to determine whether the endpoint is up to date and thus having 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 IMV the ability to request specific information about the endpoint, the protocol is not open ended, bounding the IMV to only query specific information (attributes) about specific security features (component types) of the endpoint. Each IF-M message is explicitly about a single component from the list of components in section 4.4. 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, IF-M messages contain a set of attributes which describe the particular component. Each attribute contains generic information (e.g. product information or versions) about the component so is unlikely to include any user specific or identifying information. This combination of 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 name space 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 or being re-assessed by the user's employer network than made available to the local coffee shop wireless network. While these situations do not impact the protocol itself, they do suggest that IMC 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 IF-T protocol authenticates the network's TNCS at the start of an assessment, so identity could be made available to the IMC so 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.

8 References

8.1 Normative References

[KEYWORDS] S. Bradner, "Keywords for use in RFCs to Indicate Requirement Levels", http://www.ietf.org/rfc/rfc2119.txt, IETF, March 1997.

[RFC2279] F. Yergeau, “UTF-8, a transformation format of ISO 10646”, http://www.ietf.org/rfc/rfc2279.txt, IETF, January 1998.

[RFC3339] G. Klyne, C. Newman, “Date and Time on the Internet: Timestamps”, http://www.ietf.org/rfc/rfc3339.txt, IETF, July 2002.

[RFC3986] T. Berners-Lee, R. Fielding, L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax”, http://www.ietf.org/rfc/rfc3986.txt, IETF, January 2005.

[RFC4646] Phillips, A. and Davis, M., “Tags for the Identification of Languages”, http://www.ietf.org/rfc/rfc4646.txt, September 2006.

[PA-TNC] P. Sangster, K. Narayan, “PA-TNC: A Posture Attribute Protocol (PA) Compatible with TNC”, http://www.ietf.org/rfc/rfc5792.txt, IETF, March 2010.

8.2 Informative References

[IF-ARCH] Trusted Computing Group, “TNC Architecture for Interoperability”, Specification Version 1.5, http://www.trustedcomputinggroup.org/resources/tnc_architecture_for_interoperability_specification, May 2012.

[IF-TNCCS] Trusted Computing Group, “TNC IF-TNCCS: TLV Binding”, Specification Version 2.0, http://www.trustedcomputinggroup.org/resources/tnc_iftnccs_specification, May 2014.

[IF-TNCCS-SOH] Trusted Computing Group, “TNC IF-TNCCS: Protocol Bindings for SoH”, Specification Version 1.0, http://www.trustedcomputinggroup.org/resources/tnc_iftnccs_protocol_bindings_for_soh_version_10, May 2007.

[IF-IMC] Trusted Computing Group, “TNC IF-IMC”, Specification Version 1.3, http://www.trustedcomputinggroup.org/resources/tnc_ifimc_specification, February 2013.

[IF-IMV] Trusted Computing Group, “TNC IF-IMV”, Specification Version 1.3, http://www.trustedcomputinggroup.org/resources/tnc_ifimv_specification, February 2013.

[PB-TNC] R. Sahita, S. Hanna, R. Hurst, K. Narayan, “PB-TNC: A Posture Broker Protocol (PB) Compatible with TNC”, http://www.ietf.org/rfc/rfc5793.txt, IETF, March 2010.