TCG Trusted Network Communications
TNC IF-T: Binding to TLS

Specification Version 2.0
Revision 8
27 February 2013
Published

Contact:
admin@trustedcomputinggroup.org

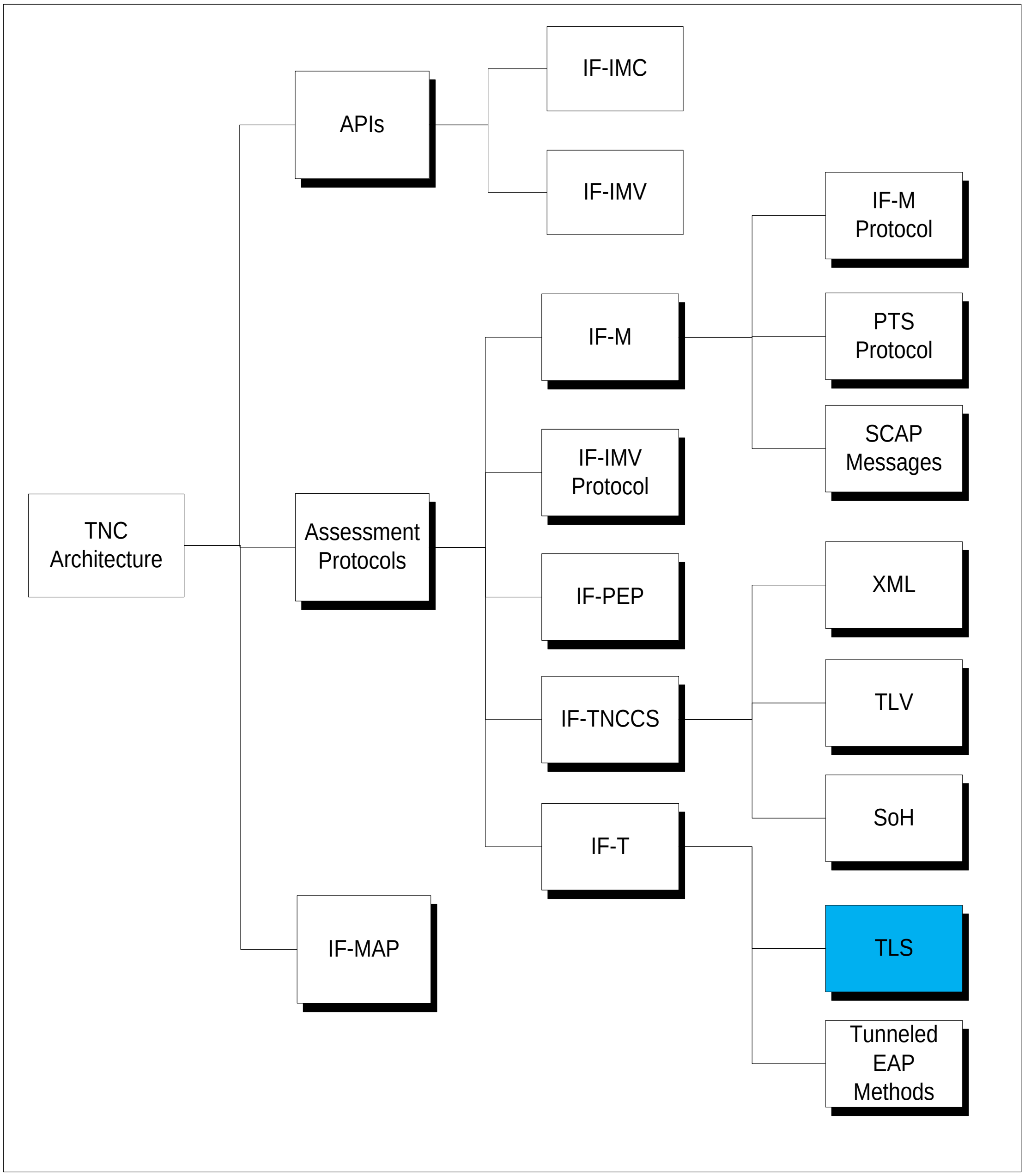
TCG PUBLISHED

TCG

Disclaimer

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.

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 TCG participants contributing to this document:

Name	Affiliation
David Louin	AMOSSYS
Scott Kelly	Aruba Networks
Padma Krishnaswamy	Battelle Memorial Institute
Jeffery Dion	Boeing
Steven Venema	Boeing
Nancy Cam-Winget	Cisco Systems
Scott Pope	Cisco Systems
Peter Wrobel	CESG
Mark Townsend	Enterasys
Ronald Marx	Fraunhofer Institute for Secure Information Technology
Sung Lee	Fujitsu Limited
Mauricio Sanchez	Hewlett-Packard
Ira McDonald	High North
Ren Lanfang	Huawei
Dr. Jiwei Wei	Huawei
Han Yin	Huawei
Stuart Bailey	Infoblox
James Tan	Infoblox
David Vigier	Infoblox
Kent Landfield	Intel Corporation
Ravi Sahita	Intel Corporation
Josh Howlett	JANET (UK)
Steve Hanna (Editor, TNC-WG co-chair)	Juniper Networks
Clifford Kahn	Juniper Networks
PJ Kirner	Juniper Networks
Lisa Lorenzin	Juniper Networks
Tom Price	Lumeta
Matt Webster	Lumeta
Atul Shah (TNC-WG co-chair)	Microsoft Corporation
Paul Sangster (Editor)	Symantec
Charles Schmidt	The MITRE Corporation
Brad Upson	University of New Hampshire InterOperability Lab
Jessica Fitzgerald-McKay	US National Security Agency
Lauren Giroux	US National Security Agency

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-T, a standard protocol used to transport the TNC assessment exchanges leveraging the existing network connectivity. Because TNC enables assessment to occur during the process of joining a network and after the endpoint has been placed on the network, several bindings of IF-T will exist to address these different scenarios.

This document defines and specifies the IF-T protocol used when the endpoint is already on the network (has an IP address) and thus able to make use of higher layer protocols such as Transport Layer Security (TLS) [TLS12] to carry the assessment. Readers interested in the use of IF-T prior to joining the network (e.g. carrying EAP message over 802.1X) should refer to the TNC IF-T: Bindings for Tunneled EAP Method specification [IF-T-EAP].

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

1.1 Interoperable with IETF PT-TLS

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-T Binding to TLS protocol to the IETF for standardization. The initial version of IF-T Binding to TLS 1.0 [IF-T-TLS1] was published long before the IETF started work on its equivalent, so in order to have alignment, version 2.0 of this specification was created and issued concurrently with the IETF's PT-TLS [PT-TLS]. It is the current intention of the TNC WG to keep the TCG IF-T Binding to TLS and IETF PT-TLS 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.

PT-EAP - IETF NEA name for the tunneled EAP method based transport protocol equivalent to the IF-T Binding for Tunneled EAP Methods specification from TCG. The PT-EAP specification was largely based upon the TCG predecessor specification.

PT-TLS - IETF NEA name for the transport protocol analogous with the protocol included in this specification. The PT-TLS specification was largely based upon the 1.0 version of this specification.

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.

2 Background

2.1 Purpose of IF-T

The IF-T protocol exists at the bottom of the TNC architecture protocol stack providing a transport service to carry the IF-TNCCS [IF-TNCCS12] [IF-TNCCS-SOH] [IF-TNCCS20] protocol over the available network. The TNC usage of IF-T enables assessments of endpoints as they are joining the network or after the endpoints are on the network. For scenarios when the endpoint is in the process of joining the network, the TNC assessment needs to be carried within the protocol used during the joining process. This protocol could be a layer two (link level) protocol, which needs to leverage an existing protocol such as 802.1 X that allows for the exchange of EAP messages. This network join-time usage is the subject of the TNC IF-T Bindings for Tunneled EAP Methods specification. This specification focuses on the IF-T usage model where the endpoint is already present on the network and thus has an IP address assigned, so is reachable using TCP/IP by other systems.

This document describes and specifies the IF-T protocol using TLS [TLS11] [TLS12]. This binding of IF-T must at least provide the same level of service as other IF-T protocol bindings. Because the endpoint is on the network and able to leverage TCP/IP, this binding of the IF-T protocol may also provide enhanced capabilities (e.g. full duplex message exchange) to IF-TNCCS in addition to potentially higher quality of service (e.g. bandwidth).

2.2 Supported Use Cases

The following IF-T use cases must be supported:

1) TNC Client initiated assessment or reassessment
   a) TNC Client becomes aware of the need to perform an assessment
   b) TNC Client uses TCP/IP to connect to the TNC Server over the network
   c) TNC Server accepts the network connection
   d) TNC Client and TNC Server exchange IF-T messages to set up a secure channel
   e) TNC Client and TNC Server exchange IF-TNCCS messages to perform the assessment
   f) TNC Client and TNC Server close the network connection

2) TNC Server initiated assessment or reassessment
   a) TNC Server becomes aware of the need to perform an assessment
   b) TNC Server uses TCP/IP to connect to the TNC Client over the network
   c) TNC Client accepts the network connection
   d) TNC Client and TNC Server exchange IF-T messages to set up a secure channel
   e) TNC Client and TNC Server exchange IF-TNCCS messages to perform the assessment
   f) TNC Client and TNC Server close the network connection

3) TNC Client establishes open connection for subsequent (TNC Client or TNC Server initiated) assessments
   a) TNC Client joins a TCP/IP network (possibly including an assessment as per use case #1)
   b) TNC Client uses TCP/IP to connect to the TNC Server over the network

c) TNC Server accepts the network connection

d) TNC Client and TNC Server exchange IF-T messages to set up a secure channel

e) TNC Client and TNC Server leave the network connection open until either decides that an assessment is necessary

f) TNC Client or TNC Server initiates an assessment

g) TNC Client and TNC Server exchange IF-TNCCS messages to perform the assessment

h) Upon completion of the assessment, the connection remains open for future use

4) TNC Client and TNC Server send IF-TNCCS messages outside of an assessment. This use case may not impact IF-T unless IF-T is aware of IF-TNCCS state (start/end of an assessment).

a) TNC Client and TNC Server already have an L3 IF-T connection left open but no active assessment

b) TNC Client and TNC Server use this session to send IF-TNCCS messages without starting an assessment (e.g. to request a SAML assertion)

c) Upon completion of this exchange, the IF-T connection remains open for future use

5) Session reuse for reassessment

a) At the end of the IF-TNCCS message exchange (e.g. steps 1d, 2d and 3f above) the TNC Client and TNC Server elect to leave open the IF-T network connection

b) Either the TNC Client or TNC Server decides to perform a reassessment using the existing open IF-T network connection

c) TNC Client and TNC Server exchange IF-TNCCS messages to perform the assessment

6) Security protected assessment

a) Prior to the IF-TNCCS message exchange of the other use cases (e.g. steps 1d, 2d, 3f and 6c above), the TNC Client or TNC Server requests the authentication of the other party

i) TNC Client may leverage a cryptographic credential or re-usable credential (password)

ii) TNC Server must use a cryptographic credential to allow for strong server authentication

b) TNC Client or TNC Server requests integrity and optionally confidentiality protection based upon byproducts of the authentication exchange.

c) TNC Client and TNC Server negotiate security protections, algorithms and keys prior to performing the IF-TNCCS message exchange

2.3 Non-supported Use Cases

The following use cases are not supported by this specification:

- Use of IF-T Binding to TLS when TCP/IP connectivity cannot be established
- Security protected assessment when no common trust anchor or cryptographic algorithms exist
- TNC Client dynamic discovery of TNC Server address after network connection (e.g. using DNS)

The following use case was supported in the IF-T Binding to TLS 1.0 and was removed from version 2.0 of this specification for compatibility with the IETF NEA PT-TLS protocol. This use case offers the ability to share the assessment connection with the application protocol, so could be supported in a future version of the IF-T protocol.

- Transport of non-TNC application data over the same TLS session as the TNC assessment. This was supported in IF-T Binding to TLS version 1.0, but was removed from 2.0 to maintain compatibility and consistency with the IETF PT-TLS specification.

2.4 Requirements

Here are the requirements that the IF-T Binding to TLS must meet in order to successfully play its role in the TNC architecture and implement the use cases listed above.

- Meets the needs of the TNC architecture

  The IF-T Binding to TLS must support all the use cases described in the TNC architecture and this specification as they apply to transporting IF-TNCCS messages between the TNCC and TNCS.

- Security

  The IF-T Binding to TLS must be capable of protecting the integrity and confidentiality of the communications between the TNC Client and TNC Server. In order to protect against impersonation and active attacks (see security considerations in section 5), the IF-T Binding to TLS must enable the TNC Client and TNC Server to strongly authenticate each other prior to the TNC assessment.

- Efficient

  The TNC architecture delays network access (or usage) until certain endpoint integrity checks have been performed. To minimize user frustration, it is essential to minimize delays and make communications using the IF-T Binding to TLS as rapid and efficient as possible. Efficiency is also important for supporting lower powered, less capable endpoint devices or when dealing with low bandwidth network connections.

- Scalable

  The IF-T binding for TLS must make it easy for the TNC Server to support many hundreds or thousands of simultaneous TNC Client connections. An idle connection should impose as little overhead as possible. This is necessary for general scaling reasons but especially because one of the use cases calls for the TNC Client to establish an open connection that may be used for subsequent assessments and leave that connection open.

- Large Data Transport

  One of the benefits of the IF-T binding for TLS is that it should be able to carry much more data than the IF-T binding for Tunneled EAP Methods, which is limited by EAP's half-duplex nature, EAP authenticator timeouts, limits on EAP message size, etc.

- Reliable

IF-T must provide reliable, in order, delivery of IF-TNCCS messages and be able to handle retransmission and fragmentation of messages if required by the underlying networking protocols.

- Full Duplex Permitted

  In order for the IF-T Binding to TLS to provide the IF-T minimal level of service, it should allow for a half duplex dialog to be transported. However, the IF-T Binding to TLS must also allow for a full duplex exchange to occur. The half duplex support provides a minimal level of message delivery service that IF-TNCCS can rely upon across IF-T bindings while support for full duplex provides a path for more robust communications when the transport allows.

- Server or Client Initiated

  The IF-T Binding to TLS must be capable of being initiated by either the TNC Client or the TNC Server.

- Extensible

  The IF-T Binding to TLS will need to be expanded over time as new features are added to the TNC architecture and new use cases identified. The IF-T Binding to TLS must be capable of being extended to provide these additions in a way that is readily recognizable by the recipient.

- Agnostic

  The IF-T Binding to TLS must not require the interpretation of the contents of the IF-TNCCS protocol data elements as part of its operation. Changes to IF-TNCCS protocol must not require the replacement of the IF-T Binding to TLS.

2.5 Non-Requirements

Here are certain requirements that the IF-T Binding to TLS is not required to meet.

- Use Prior to Network Connectivity

  The IF-T Binding to TLS is not expected to be usable prior to the TNC Client possessing an IP address, routes and other information enabling it to have IP layer access to the network. For situations where the TNC Client is not yet present on the network, the IF-T binding for Tunneled EAP Methods should be used.

2.6 Assumptions

Here are the assumptions that this specification makes about the network connectivity available to the TNC Client. This assumption differs from the expectations for L2-only connectivity used by the prior IF-T Binding for Tunneled EAP methods.

- TCP/IP Connectivity

  Prior to the use of the IF-T Binding to TLS, the TNC Client and TNC Server are both able to communicate with each other over TCP/IP. This communication may be limited to the communication path between the TNC Client and TNC Server recognizing that the endpoint might only be able to reach a very small number of systems on the network during the assessment.

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 Network Communications Diagram Conventions

This specification includes diagrams illustrating the format and contents of network messages exchanged between the Network Access Requestor (NAR) and Network Access Authority (NAA). These diagrams depict the 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-byte fields representing numeric values must be sent in network (big endian) byte order. The values of each bit field (e.g. flags) 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 single byte field with only bit location 0 set has the value 0x80.

3 Network Connected Endpoint Assessments

This document specifies the IF-T binding for use when performing an assessment or reassessment after the endpoint has been admitted to the network and is capable of using TCP/IP to communicate with the TNC Server. If the endpoint does not yet have TCP/IP layer access to the TNC Server (and vice versa), the endpoint should use the IF-T Binding for Tunneled EAP Methods when performing an assessment.

Because the endpoint has TCP/IP access to the TNC Server (potentially on a restricted portion of the network), the TNC Client and TNC Server have the ability to establish (or re-use) a reliable TCP/IP connection in order to perform the assessment. The TCP/IP connection enables the assessment to occur over a relatively high performance, reliable channel capable of supporting multiple roundtrip message exchanges in full duplex manner. These connection properties are very different from what is available when the endpoint is initially joining the network (e.g. during an 802.1X based assessment), therefore the design described in this specification follows a different path to maximize the benefits of the connection properties.

3.1 Benefits

This binding of IF-T is normally able to offer to the TNC Client and TNC Server significantly higher quality of service and flexibility of operation than other bindings. However, there may be some added risks when the endpoint is on the network prior to its initial assessment (if no admission time assessment is performed). Because of these risks, the combined use of an EAP-based assessment during admission followed by reassessment using TCP/IP may be appropriate in many environments.

Some of the benefits to having a TCP/IP based transport during an assessment include:

- Full Duplex connectivity – can send multiple assessment messages prior to receiving a response including sending of asynchronous messages (e.g. alerts of posture changes)
- High Bandwidth – potentially much higher bandwidth than other transports (e.g. 802.1X) allowing more in-band data (e.g. remediation, verbose posture information)
- Reliability – IF-T messages sent will not be lost in transit since they are acknowledged by underlying TCP/IP protocol
- In-order Delivery – IF-T messages can be sent knowing they won't be received prior to earlier messages
- Large Messages – ability to send very large IF-M messages without directly fragmenting them (underlying carrier protocol may introduce fragmentation)
- Bi-directional – TNC Client and TNC Server can initiate an assessment or reassessment
- Multiple Roundtrips – TNC Client and TNC Server can exchange numerous messages without fear of infrastructure timeouts. However, the entire exchange should be kept as brief as possible in case the user has to wait for its completion.

In order to take full advantage of the above listed benefits, the IF-T binding in this specification does not re-use existing IF-T technologies (e.g. EAP and EAP-TNC). However, this IF-T binding must still meet the same core set of IF-T requirements (e.g. security) in order for IF-TNCCS to be able to operate over both types of transports, but may provide additional or higher qualities of service. See the Security Considerations section for details of how these requirements are met.

3.2 Securing the TCP/IP Session with TLS

All bindings of IF-T must be capable of providing strong authentication, integrity and confidentiality protection for the IF-TNCCS messages. Rather than define a new protocol over TCP/IP to provide adequate protection, this specification requires the use of Transport Layer Security [TLS12] to secure the connection. TLS was selected because it's a widely deployed protocol with parallel protections to a number of the tunneled EAP methods, and it meets most of the security requirements (this specification will describe additional security protections offered in section 4.5). Therefore, the remainder of this specification will describe the use of IF-T on top of the TLS protocol.

3.3 No Change to Base TLS Protocol

During the design of the IF-T Binding to TLS protocol, several approaches were considered with different costs and benefits. Several of these approaches involved integrating the IF-T protocol into the TLS handshake protocol. Because the IF-T protocol requires the underlying TLS carrier to provide security protections, the IF-T protocol couldn't operate before the cipher suites were negotiated and in use. One option was to integrate into the TLS handshake protocol after the ChangeCipherSpec phase allowing the IF-T message to be protected. The benefit of this approach is that the assessment protocol could operate below the application protocols allowing for easier integration into applications. However, making this change would require some extensions to the TLS handshake protocol standards and existing widely deployed TLS implementations, so it wasn't clear that the cost was warranted, particularly because the application independence can also be offered by a shim library between the application and TLS library that provides the PT protocol encapsulation/decapsulation.

The other general approach considered was to have IF-T layer on top of TLS as an application protocol (using the standard application_data ContentType). This has the advantage that existing TLS software could be used. However, the IF-TNCCS traffic would need to be encapsulated/decapsulated by a new protocol layer before being passed to the TLS library. This didn't seem like a significant issue as IF-TNCCS is architected to layer on IF-T protocol anyway.

After considering the different options, it was determined that layering the IF-T protocol on top of the TLS protocol without requiring current TLS protocol implementations to change met all the requirements and offered the best path toward rapid adoption and deployment. Therefore the following sections describe the IF-T Binding to TLS protocol which is carried on top of TLS.

3.4 Parallel Enumerated Values

The IF-T Binding to TLS and the equivalent PT-TLS protocol have several fields that contain enumerated fields defined in the IF-T Binding to TLS or PT-TLS standards. These values need to be the same to achieve interoperability between TNC-based and IETF-based implementations. In order to provide interoperability in the standard namespaces while allowing for parallel vendor-defined namespaces for other uses, IF-T Binding to TLS includes a namespace identifier immediately prior to the field capable of containing a value from multiple namespaces.

It is also important that each of the field's namespaces be readily extensible without constant coordination yet 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 specifications could augment. For example, the IETF's IANA maintains a set of values standardized within the IETF. To maximize interoperability and avoid duplicating values defined in the IETF namespace, this specification references the IETF IANA defined values and uses them in compatible ways.

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 PT-TLS 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 PT-TLS specification to interoperate with TNC IF-T Binding to TLS implementations while still allowing TCG to have additional capabilities (e.g. for TPM integration).

4 IF-T Over TLS Protocol

This section specifies the IF-T transport protocol used on top of TLS. This protocol runs directly on top of TLS as an application. This means IF-T is encapsulated within the TLS Record Layer protocol using the standard ContentType for applications (application_data).

4.1 TCP Port Usage

In order for an assessment initiator to establish a TCP connection to its peer, the initiator needs to know the TCP port number on which the recipient is listening for assessment requests. Note that for support of all of the above listed use cases, both TNC Client and TNC Server need to be capable of listening for requested assessments. The IETF has reserved the well known TCP port number 271 for the PT-TLS protocol for use as a listening port for software willing to accept new inbound PT-TLS and thus IF-T Binding to TLS connections.

4.2 Preventing MITM Attacks with Channel Bindings

As described in the NEA Asokan Attack Analysis [ASOKAN], a sophisticated MITM attack can be mounted against NEA or TNC systems. The attacker forwards IF-M (or PA-TNC) messages from a healthy machine through an unhealthy one so that the unhealthy machine can gain network access. Because there are easier attacks on NEA systems, like having the unhealthy machine lie about its configuration, this attack is generally only mounted against machines with an External Measurement Agent (EMA). The EMA is a separate entity, difficult to compromise, which measures and attests to the configuration of the endpoint. For TCG-based trusted platforms, the EMA could be the PTS leveraging a TPM to provide a signed Integrity Report for the system.

To protect against NEA Asokan attacks, the TNC Client on a platform including the PTS (or equivalent) SHOULD pass the tls-unique channel binding [BINDINGS] for IF-T's underlying TLS session to the PTS for inclusion in TPM-based quote operations. This value can then be included in the PTS's attestation and the IMV responsible for communicating with the PTS may then confirm that the value matches the tls-unique channel binding for its end of the connection. If the values match, the posture sent by the PTS and NEA Client is from the same endpoint as the client side of the TLS connection (since the endpoint knows the tls-unique value), so no man-in-the-middle is forwarding posture. If they differ, an Asokan attack has been detected. The IMV MUST fail its verification of the endpoint if an Asokan attack has been detected

4.3 IF-T Message Flow

This section discusses the general flow of messages between the TNC Client's Network Access Requestor and the TNC Server's Network Access Authority in order to provide an assessment using the IF-T Binding to TLS. This section does not discuss the underlying message exchanges used by TCP and TLS, instead focusing on the IF-T messages.

4.3.1 Cause of an Assessment

Initially, the TNC Client or TNC Server will decide that an assessment is needed. What stimulates the decision to perform an assessment is outside the scope of this specification, but some examples include:

- TNC Server becoming aware of suspicious behavior by an endpoint
- TNC Server receiving new policies requiring immediate action
- TNC Client noticing a change in local security posture
- TNC Client wishing to access a protected network or resource

Because either the TNC Client or TNC Server can trigger the establishment of the TLS session and initiate the assessment, this document uses the terms "assessment initiator" when referring to the party which initiated the assessment. Similarly, this specification uses the term "assessment responder" for the party which is listening and accepting the IF-T Binding to TLS assessment session. This nomenclature allows either TNC component to fill either of the IF-T roles.

4.3.2 Issues with Server Initiated TLS Sessions

The IF-T Binding to TLS allows for either the TNC Client or TNC Server to establish the TLS session. However, there are several potential issues associated with having the TNC Server establish the TLS session to the TNC Client. Allowing the TNC Server to establish the TLS connection means that TNC Clients will need to be listening for a connection request on a TCP port known by the TNC Server. In many deployments, the security policies (e.g. host-based firewall) of an endpoint are designed to minimize the number of open inbound TCP/UDP ports that are available to the network to reduce the potential attack footprint. When the TNC Server initiates a TLS session to the TNC Client, the TNC Client is effectively acting as the TLS server during the protocol exchange. This means the TNC Client would need to possess an X.509 certificate to protect the initial portion of the TLS handshake. In situations where the TNC Server initiates the creation of the TLS session, both the TNC Client and TNC Server MUST possess and use X.509 certificates to fully authenticate the session. For many deployments, provisioning X.509 certificates to all TNC Clients has scalability and cost issues; therefore, it is recommended that the TNC Client not listen for connection requests from the TNC Server but instead establish and maintain a TLS session to the TNC Server proactively so either party can initiate an assessment using the preexisting TLS session as required.

Another issue with the NEA Server acting as the TLS client involves certificate path validation. In this case, the NEA Server presents its certificate (also used when it is acting in the TLS server role) during the client authentication. In this situation, the TNC Client (acting as the TLS server) will need to follow the certificate path validation rules as defined in RFC 5280 [RFC5280]. Both the TNC Client and Server also need to be able to authorize the session by matching the Subject and SubjectAltName fields for certificates issued by a particular trusted certificate issuer.

Therefore, TNC Clients SHOULD be capable of establishing and holding open a TLS session with the TNC Server immediately after obtaining network access. TNC Client MAY allow for the TNC Server to establish a new TLS session when one does not already exist. Having an existing TLS session allows either party to initiate an assessment without requiring the TNC Client to be listening for new connection requests. In order to keep the TLS session alive, the TNC Client and TNC Server SHOULD be capable of supporting the TLS heartbeat protocol [HEARTBEAT].

4.3.3 Establish or Re-Use TLS Session

If the assessment initiator already has TLS connectivity to the assessment responder, the assessment initiator may re-use the session otherwise a new TLS session is required. Note that an existing TLS session between the NAR and NAA may be used to start an assessment regardless of which component originally established the session.

4.3.4 IF-T Message Exchange

The IF-T Binding to TLS message exchange occurs in three distinct phases:

- TLS Setup (includes TLS Handshake protocol)
- IF-T Negotiation
- IF-T Data Transport

The TLS Setup phase is responsible for the establishment of the TCP connection and the TLS protections for the IF-T messages. The TLS Setup phase starts with the establishment of a TCP connection between the NAR and NAA. The new connection triggers the TLS Handshake protocol to establish the cryptographic protections for the TLS session. The TLS Setup phase MUST NOT be repeated after the IF-T Data Transport phase has been reached unless a change of TLS cipher suite or keying material is required to properly protect the session.

The IF-T Negotiation phase is only performed at the start of the first assessment on a TLS session. During this phase, the NAR and NAA discover each other's IF-T capabilities and establish a context that will apply to all future IF-T messages sent over the TLS session. The IF-T Negotation phase MUST NOT be repeated after the session has entered the IF-T Data Transport phase. TNC assessment (IF-TNCCS) messages MUST NOT be sent by the NAR or NAA prior to the completion of the IF-T Negotiation phase when the security protections for the session are established and applied to the messages.

Finally the IF-T Data Transport phase allows the NAR and NAA to exchange IF-T messages under the protection of the TLS session and consistent with the capabilities established in earlier phases. The exchanged messages can be an IF-T protected assessment as described in this specification or other TNC Client/TNC Server exchanged messages.

4.3.4.1 TLS Setup Phase

After a new TCP connection is established between the NAR and NAA, a standard TLS exchange is performed to negotiate a common security context for protecting subsequent communications. As discussed in section 4.3.2, the TCP connection establishment and/or the TLS handshake protocol could be initiated by either the TNC Client or TNC Server. The most common situation would be for the assessment initiator to trigger the creation of the TCP connection and TLS handshake, so an assessment could begin when no session already exists. When the TNC Server has initiated the TLS Setup, the TNC Server is acting as a TLS client and the TNC Client is the TLS server (accepting the inbound TLS session request). The expected normal case is that the TNC Client initiates this phase, so that the TNC Server is acting as the TLS server and therefore the bootstrapping of the security of the TLS session is using the TNC Server's certificate. Having the TNC Client initiate the TLS session avoids the need for the TNC Client to also possess a certificate.

During this phase the initiator of the TLS session (normally the TNC Client) contacts the listening port of the other party. The IF-T Binding to TLS assessment responder MUST use an X.509 certificate when authenticating to the assessment initiator to bootstrap the security protections of the TLS session. The IF-T Binding to TLS assessment initiator MAY also use an X.509 certificate as a TLS client authenticator providing for a bi-directional authentication of the TLS session. The TNC Client MUST provide RFC 5280 [CRL] compliant certificate validation when evaluating the server certificate. The TNC Client MAY perform certificate revocation checking on the TNC Server's certificate. Several forms of certificate validation are defined, so IF-T Binding to TLS allows the TNC Client to decide on what certificate revocation technique is to be used. Note that in order for the TNC Client to perform certificate validation, some network access (e.g. HTTP) might need to be allowed during the TLS handshake.

Similarly, the TNC Client MUST perform a RFC 6125 [NAME-VALID] compliant TNC Server domain name validation against the contents of the server certificate factoring in the following restrictions:

o Any SRV-IDs and URI-IDs present in the certificate are ignored

o CN-IDs SHOULD NOT be present in the certificates

o Wildcards MUST NOT appear in the DNS-ID or CN-ID of a certificate identifying a PT-TLS Server.

Details for the reverse direction are given in section 4.3.2.

TNC Client implementations of this specification integrated with a TCG trusted platform environment SHOULD be capable of using a client side X.509 certificate including the Subject Key Attestation Evidence (SKAE) extension [SKAE] for client authentication during the TLS handshake. The SKAE extension includes evidence that the private key associated with the public key found in the certificate is resident inside a TPM. The use of a TPM resident private key during the establishment of a TLS session provides a strong binding between a particular TPM on the TLS session initiator (TNC Client) and the TLS session being established. The TNC Server SHOULD process the certificate as usual and additionally performs a validation of the SKAE's evidence using the signing AIK private key. After the TLS session has been successfully created using a certificate containing the SKAE extension, the TNC Server SHOULD be capable of requesting an attestation using an Integrity Report [INTREPORT] from the PTS [IF-PTS] on the TNC Client leveraging the TPM resident key. The attestation would occur during the IF-T Data Transport phase using an IMV supporting the PTS information. The TNC Server SHOULD verify that the authentication credentials are associated with the same TPM as the one used for the PTS exchange. The strong cryptographic binding between the TNC Client's TLS identity and TPM resident key during the TLS handshake with the use of the TPM resident key during a subsequent attestation provides a countermeasure to MITM attack described in section 5. The active MITM is unable to both act as: the TNC Client (requesting network access) performing the TLS handshake using the certificate with the SKAE evidence and also having access to TPM resident keys on another clean system. Therefore the TNC Server can detect when a different system is providing the attestation information than the system that performed the TLS handshake.

Due to deployment issues with issuing and distributing certificates to a potentially large number of TNC Clients, this specification allows the TNC Client to be authenticated during the IF-T Negotation phase using other more cost effective methods. At the conclusion of a successful initial TLS Setup phase, the NAR and NAA have a protected session to exchange messages. This allows the protocol to transition to the IF-T Negotiation phase.

4.3.4.2 IF-T Negotiation Phase

Once a TLS session has been established between NAR and NAA, the assessment initiator sends a Version Request Message indicating its supported IF-T protocol version range. Next the assessment responder sends a Version Response Message which selects a protocol version from within the range offered. The assessment responder SHOULD select the preferred version offered if supported otherwise the highest version that it is able to support from the received Version Request Message. If the assessment responder is unable or unwilling to support any of the versions included in the Version Request Message, the responder SHOULD send an IETF Version Not Supported error code in an IF-T Error message.

If no client side authentication has occurred during the TLS Setup phase, the NAA can authenticate the client using IF-T client authentication messages as described in 4.7. The NAA initiates the client authentication and indicates when the authentication is complete.

When the NAR receives the SASL [SASL] Mechanisms list, the TNC Client responds with a SASL Mechanism Selection message indicating the method of authentication to be used. Upon selecting an appropriate SASL mechanism, the NAA and NAR exchange SASL mechanism specific messages in order to authenticate the TNC Client. When the client authentication successfully completes and no additional authentications are required (as indicated by the NAR sending an empty SASL Mechanisms list), the IF-T for TLS session transitions into the IF-T Data Transport phase, where it will remain for the duration of the session. Note that the NAR could choose to not authenticate the client (indicated by only sending an empty SASL Mechanisms list) or to continue performing a posture assessment even if the authentication did not complete successfully.

4.3.4.3 IF-T Data Transport Phase

Once an IF-T session is available to carry IF-TNCCS based assessments, the IF-T Binding to TLS allows either the NAA or NAR to start an assessment when provided an IF-TNCCS message for transmission. The IF-TNCCS 2.0 standard prescribes whether the TNC Client or TNC Server starts the assessment. The assessment initiator envelopes the IF-TNCCS message in an IF-T message, assigning a message identifier to the message and sending it over the session. The assessment recipient validates the IF-T message and delivers the encapsulated IF-TNCCS message to its upstream component (TNC Client or TNC Server).

Most IF-T messages contain IF-TNCCS messages that request posture information or a response containing the requested information. The NAR and NAA may also exchange messages between them, such as an IF-T Error Message indicating that a problem occurred processing a message. During an assessment, the NAR and NAA merely encapsulate and exchange the IF-TNCCS messages and are unaware of the state of the assessment. The IF-T Binding to TLS allows either party to send an IF-T message at any time reflecting the full duplex nature of the underlying TLS session. For example, an assessment initiator may send several IF-TNCCS messages prior to receiving any responses from the peer assessment responder. All implementations of the IF-T Binding to TLS MUST support full duplex IF-T message exchange. However, some IF-TNCCS protocols may not be able to make use of the full-duplex message exchange.

4.3.5 TLS Requirements

In order to ensure that strong security is always available for deployers and to improve interoperability, this section discusses some requirements on the underlying TLS transport used by IF-T.

TLS is a popular security protocol with active research and protocol evolution. As of the 2.0 version of this specification, there are three versions of TLS (1.0-1.2) deployed to varying degrees. This specification encourages the use of the latest version of TLS whenever possible, but recognizes that the latest version might not be widely implemented or deployed immediately after the TLS version is defined. IF-T Binding to TLS implementations SHOULD support the latest standardized version of TLS which currently is TLS 1.2 [TLS12]. However, TLS 1.2 implementations are not currently widely adopted, so implementations might achieve interoperability more quickly by initially supporting TLS 1.1 until TLS 1.2 has been more widely adopted.

For each TLS version supported, implementations of the IF-T Binding to TLS MUST at least support the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite. This cipher suite requires the server to provide a certificate that can be used during the key exchange. Implementations SHOULD NOT include the support for cipher suites that do not minimally offer NAA authentication, such as the anonymous Diffie-Hellman cipher suites (e.g. TLS_DH_anon_WITH_AES_128_CBC_SHA). Implementations MUST support RFC 5746 [TLS-RENEGO]. Implementations MAY allow renegotiation to provide confidentiality for the client certificate. If renegotiation is allowed implementations need to select the appropriate handshake messages as described in RFC 5929 [BINDING-TLS] for the tls-unique value. After the TLS Setup Phase completes, TLS renegotiation is no longer allowed during the session.

4.4 IF-T Message Format

This section describes the format and semantics of the IF-T Binding to TLS message. Every IF-T Binding to TLS compliant message MUST start with the IF-T header described in this section. The IF-T header provides a simple Type-Length-Value (TLV) based envelope around the IF-T message payload such as an IF-TNCCS message batch. Note that the Reserved and Message Identifier fields are technically part of the value portion of the TLV. However because these fields are required to be present in every IF-T message, they are described here as preceding the variant part (Message Value field) of the message.

The following is the TLV-based protocol for IF-T:

```
                     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   |           Message Type Vendor ID              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Message Type                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Message Length                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Message Identifier                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Message Value (e.g. IF-TNCCS Message) . . . .       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Reserved	This field MUST be set to 0 upon transmission and MUST be ignored by compliant IF-T message recipient implementations.
Message Type Vendor ID	This field indicates the owner of the name space associated with the Message Type. This is accomplished by specifying the 24 bit SMI Private Enterprise Number Vendor ID of the party who owns the Message Type name space. TCG unique (not in IETF NEA's specification) standard messages defined in this specification MUST use the TCG SMI Private Enterprise Number value (0x005597) in this field. Values shared with the IETF MUST use the IETF SMI Private Enterprise Number value (0) in this field.
Message Type	This field defines the type of the IF-T message (within the scope of the specified vendor name space included in the Message Value Vendor ID field). Recipients of a message containing a vendor id and message type that is unrecognized SHOULD respond with an IETF NEA Type Not Supported error code in an IF-T Binding to TLS Error message. NAA and NAR MUST NOT require support for particular vendor-defined IF-T Message Types and MUST interoperate with other parties despite any differences in the set of vendor-defined IF-T Message Types supported (although they MAY permit administrators to configure them to require support for specific vendor-defined IF-T message types). The Message Type value of 0xffffffff is reserved. NAA and NAR MUST NOT send IF-T messages in which the IF-T Message Type has this reserved value (0xffffffff). If an NAA or NAR receives a message in which the Message Type has this reserved value (0xffffffff), it SHOULD respond with an IETF NEA Invalid Parameter error code in an IF-T Binding to TLS Error message.
Message Length	This field contains the length in octets of the entire IF-T message (including the entire header). Therefore, this value MUST always be at least 16. Any NAA and NAR that receives a message with a Message Length field whose value is less than 16 SHOULD respond with an IETF NEA Invalid Parameter in an IF-T Error message. Similarly, if a NAA or NAR receives an IF-T message for a Message Type that has a known Message Length and the Message Length indicates a different value (greater or less than the expected value), the recipient SHOULD respond with an IETF NEA Invalid Parameter error code in an IF-T Binding to TLS Error message.
Message Identifier	This field contains a value that uniquely identifies the IF-T message on a per message sender (NAR or NAA) basis. This value can be copied into the body of a response message to indicate which message was received and caused the response. For example,

	this field is included in the IF-T Error Message so the recipient can determine which message sent caused the error. The Message Identifier MUST be a monotonically increasing counter starting at zero indicating the number of the messages the sender has transmitted over the TLS session. It is possible that a busy or long lived session might exceed 2^{32}-1 messages sent, so the message sender MUST roll over to zero upon reaching the 2^{32}nd message, thus restarting the increasing counter. During a rollover, it is feasible that the message recipient could be confused if it keeps track of every previously received Message Identifier, so recipients MUST be able to handle roll over situations without generating errors.
Message Value	The contents of this field vary depending on the particular Message Type being expressed. This field normally contains an IF-TNCCS message.

4.5 IF-T Message Types

This section defines the TNC standard IF-T Message Types used to carry IF-T related and IF-TNCCS messages between the NAR and NAA. The following table summarizes the message type values that are used when the Vendor ID is set to the TCG SMI PEN (0x005597).

Message Type Name	**TNC Standard Message Type**	**Description**
Experimental	0	Reserved for use in specification examples, experimentation and testing. This message type MUST only be sent when the TNC Client and TNC Server are in the IF-T Data Transport phase and only on a restricted, experimental network. Production code MUST send an Invalid Message error code in the IF-T Error message if an experimental message is received.
Reserved	1-8	These values are reserved for future use and were allocated in version 1.0 of the IF-T Binding to TLS protocol but these values have been migrated to the IETF namespace so will not be used for version 2.0 of this specification to avoid confusion.

IFT_TNCCS_SOH_10_BATCH	9	Contains an IF-TNCCS SOH message. For more information on IF-TNCCS binding to SoH messages see the IF-TNCCS: Protocol Bindings for SoH specification [IF-TNCCS-SOH].
IFT_TNCCS_XML_10_BATCH	10	Contains an XML-based IF-TNCCS 1.x (1.0, 1.1 or 1.2) message. For more information on IF-TNCCS see the IF-TNCCS specification [IF-TNCCS12].

Note that this table is not as long as it was in IF-T Binding to TLS 1.0. The IETF NEA working group has adopted the majority of the Message Types defined in version 1.0 of this specification, so rather than duplicate those values in the TNC namespace this specification references the IETF namespace. The few Message Types that remain defined in the TCG name space allow for carrying of earlier versions of the IF-TNCCS protocol such as the XML and SoH variations of IF-TNCCS. The Message Types supported in the IETF NEA spec include: IF-T version selection messages, SASL client authentication messages, and an IF-T protocol error message.

Implementations of IF-T Binding to TLS version 2.0 primarily use the message types defined within the IETF NEA namespace in order to achieve interoperability while using the types in the TNC name space to carry legacy TNC protocols. Implementations supporting only version 2 of the IF-T Binding to TLS MUST NOT send the message types (1-8) defined in the TCG namespace from IF-T Binding to TLS version 1.0. Implementations supporting both version 1.0 and version 2.0 of the IF-T Binding to TLS will need to detect what versions are supported by the other party involved in an assessment. This needs to be done with care to avoid breaking compatibility with version 2.0 (and IETF NEA) only implementations. See section 4.6 for more information on version negotiation.

IF-T Binding to TLS version 2.0 based assessments MUST NOT include version 1.0 message types 1-8 as defined in the TCG namespace as these values are now reserved and would not be understood by an IETF PT-TLS implementation. Client authentication in version 2.0 is performed using messages from the IETF namespace and is based upon a different authentication technology (SASL). TNC implementations wishing to implement IF-T Binding to TLS version 2.0 MUST use the Message Type values defined in the IETF Standard PT-TLS Message Types section of PT-TLS specification with an IF-T Message Type 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. It is envisioned that future TNC specifications will assign values from the TCG namespace.

The following table shows the IETF NEA defined Message Types that are to be used with the IETF's SMI PEN (0x000000). For an up to date list of IETF NEA defined Message Types refer to the IETF IANA repository for PT-TLS Message Types.

Message Type Name	**NEA Standard Message Type**	**Description**
Experimental	0	Reserved for use in specification examples, experimentation and

		testing. This message type MUST only be sent when the TNC Client and TNC Server are in the IF-T Data Transport phase and only on a restricted, experimental network. Production code MUST send an Invalid Message error code in the IF-T Error message if an experimental message is received.
Version Request	1	Version negotiation request including the range of versions supported by the sender. This message type MUST only be sent by the IF-T assessment initiator as the first IF-T Binding to TLS message in the IF-T Negotiation phase. Recipients MUST send an Invalid Message error code in an IF-T Binding to TLS Error message if a Version Request is received at another time.
Version Response	2	IF-T Binding to TLS protocol version selected by the assessment responder. This message type MUST only be sent by the IF-T assessment responder as the second message in the IF-T Negotiation phase. Recipients MUST send an Invalid Message error code in an IF-T Error message if a Version Response is received at another time.
SASL Mechanisms	3	Sent by the TNC Server to indicate what SASL mechanisms it is willing to use for authentication on this session. This message type MUST only be sent by the TNC Server in the IF-T Negotiation phase. The TNC Client MUST send an IETF NEA Invalid Message error code in a IF-T Error message if a SASL Mechanisms message is received at another time. An empty SASL Mechanisms list indicates the end of the client authentication exchange.
SASL Mechanism Selection	4	Sent by the TNC Client to select a SASL mechanism from the list offered by the TNC Server. This message type MUST only be sent by the TNC Client in the IF-T Negotiation phase. The TNC Server

		MUST send an IETF NEA Invalid Message error code in a IF-T Error message if a SASL Mechanism Selection is received after the IF-T Negotiation phase. Once a SASL mechanism has been selected, it may not change until the mechanism completes either successfully or as a failure.
SASL Authentication Data	5	Opaque octets exchanged between the TNC Client and TNC Server's SASL mechanisms to perform the client authentication. This message type MUST only be sent during the IF-T Negotiation phase. Recipients MUST send an IETF NEA Invalid Message error code in a IF-T Error message if a SASL Authentication Data message is received after the IF-T Negotiation phase.
SASL Result	6	Indicates the result code of the SASL mechanism authentication. This message type MUST only be sent by the TNC Server when the TNC Client and TNC Server are in the IF-T Negotiation phase. The TNC Client MUST send an IETF NEA Invalid Message error code in a IF-T Error message if a SASL Result is received after the IF-T Negotiation phase.
PB-TNC (IF-TNCCS) Batch	7	Contains an IF-TNCCS 2.0 batch. This message type MUST only be sent when the TNC Client and TNC Server are in the IF-T Data Transport phase. Recipients SHOULD send an IETF NEA Invalid Message error code in a IF-T Error message if a IF-TNCCS Batch is received outside of the IF-T Data Transport phase.
PT-TLS (IF-T) Error	8	IF-T Binding to TLS Error message as described in section 4.8. This message type may be used during any IF-T phase.
Reserved	9+	These values are reserved for future allocation by the IANA. Recipients of messages of type 9 or higher that do not support the IF-T Binding to TLS Message Type

		Vendor ID and Message Type MUST respond with an IETF Type Not Supported error code in an IF-T Error message.

4.6 IF-T Version Negotiation

This section describes the message format and semantics for version 2.0 of the IF-T Binding to TLS version negotiation. Version 1.0 of the IF-T Binding to TLS supported a Version Request and Version Response pair of messages that use the TCG's Message Type Vendor ID value of 0x005597 and Message Types 1 and 2. Version 2.0 of the IF-T Binding to TLS includes an equivalent Version Request and Version Response pair of messages, but these are differentiated by recipients as they use the IETF's Message Type Vendor ID value of 0 and Message Types 1 and 2.

In order to support backward compatibility yet emphasize interoperability with the IETF NEA's PT-TLS protocol, implementations supporting both version 1.0 and version 2.0 SHOULD initiate the version negotiation exchange using version 2.0 messages unless the implementation already knows the other party only supports version 1.0 (e.g. via configuration or prior interactions). Version 1.0 only implementations will respond with an IETF Type Not Supported Error Code in an IF-T Error message when receiving a version 2.0 Version Request message indicating that version 2.0 is not supported so version 1.0 should be tried next.

The message types described in this sub-section allow the initiator of an IF-T session to trigger a version negotiation at the start of an assessment. The IF-T assessment initiator MUST send a Version Request message as its first IF-T message and MUST NOT send any other IF-T messages on this connection until it receives a Version Response message or an Error message. The IF-T session responder MUST complete the version negotiation (or respond with an Error message) prior to sending or accepting reception of any additional messages. After the successful completion of the version negotiation, both the NAA and NAR MUST only send messages compliant with the negotiated protocol version. Subsequent assessments on the same session MUST use the negotiated version number and therefore MUST NOT send additional version negotiation messages.

The remainder of this sub-section describes version 2.0 of the Version Negotiation exchange messages that are sent using the IETF's Message Type Vendor ID. Note that the syntax and semantics are nearly identical to version 1.0 exchange that are sent using the TCG's Message Type Vendor ID, so implementations may be able to re-use substantial amounts of code.

4.6.1 Version Request Message

This message is sent at the start of the first assessment on an assessment session between the NAA and NAR. This message contains the sender's supported versions of the IF-T Binding to TLS protocol. Recipients of this message MUST respond with a Version Response or an IF-T Binding to TLS Error message containing the IETF's Version Not Supported or Invalid Message 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   |    Min Vers   |    Max Vers   |   Pref Vers   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Reserved	This field MUST be set to 0 upon sending and MUST be ignored by compliant recipients.

Min Vers	This field contains the minimum version of the IF-T Binding to TLS protocol supported by the sender. This field MUST be set to 1. Note that even though this is version 2 of the IF-T Binding to TLS protocol, a 1 is used in this field in order to maintain compatibility with IETF NEA PT-TLS since it's the first version of this attribute used within the IETF's Vendor ID namespace.
Max Vers	This field contains the maximum version of the IF-T Binding to TLS protocol supported by the sender. This field MUST be set to 1. Note that even though this is version 2 of the IF-T Binding to TLS protocol, a 1 is used in this field in order to maintain compatibility with IETF NEA PT-TLS since it's the first version of this attribute used within the IETF's Vendor ID namespace. However, future versions of this specification will probably remove this requirement so IF-T Binding to TLS assessment responders MUST be prepared to receive other values.
Pref Vers	This field contains the sender's preferred version of the IF-T Binding to TLS protocol. This is a hint to the recipient that the sender would like this version selected if supported. The value of this field MUST fall within the range of Min Vers to Max Vers. This field MUST be set to 1 to align with the IETF NEA PT-TLS protocol. However, future versions of this specification will probably remove this requirement so IF-T Binding to TLS assessment responders MUST be prepared to receive other values.

4.6.2 Version Response Message

This message is sent in response to receiving a Version Request Message at the start of a new assessment session. If a recipient receives a Version Request after a successful version negotiation has occurred on the session, the recipient SHOULD send an Invalid Message error code in a IF-T Error message and have TLS cleanly close the session.

```
                     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                    |    Version    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Reserved	This field MUST be set to 0 upon sending and MUST be ignored by compliant recipients.
Version	This field contains the version selected by the sender of this message. The version selected MUST be within the Min Vers to Max Vers inclusive range sent in the Version Request Message. If an IF-T assessment initiator receives a message with an invalid Version selected, the IF-T assessment initiator MUST respond with an IETF NEA Version Not Supported error code in an IF-T Error message.

4.7 Client Authentication using SASL

This section includes a description of the message format and semantics necessary to perform client authentication (authentication of the TNC Client) over the IF-T Binding to TLS. Client authentication could be necessary if the TNC Server requires such an authentication and it was not performed during the TLS Setup phase (TLS handshake).

Version 1.0 of the IF-T Binding to TLS included a simple client authentication framework which was replaced in version 2.0 with the SASL framework. The general model used for performing an authentication of the client using IF-T Binding to TLS version 2.0 uses the Simple Authentication and Security Layer (SASL) [SASL] framework and leverages its many standard mechanisms. SASL provides a number of standards-based authentication mechanisms capable of authenticating the TNS Client using a variety of base technologies.

Client authentication may occur during the TLS handshake using TLS defined authentication techniques. Because client authentication is optional for TLS, the TNC Server's policy may require the client to be authenticated by IF-T before performing the assessment. Similarly, the TNC Server may require an IF-T authentication even if the TNC Client was authenticated during the TLS handshake (e.g. to allow a user authentication after a system level authentication occurred during the TLS handshake). The decision of whether a SASL client authentication is required is left to the TNC Server's policy.

As discussed in section 4.3.2, it is possible that the TNC Server may initiate the TLS session to the TNC Client, thus causing the TNC Server to fill the role of TLS Client during the TLS handshake. Because the TNC Server is required to possess an X.509 certificate for use when it is acting as the TLS Server role (normal case), IF-T requires that the TNC Server MUST use its X.509 certificate for TLS client authentication during the TLS handshake even when it is acting as the TLS Client. In this case, the NEA Client and NEA Server will authenticate using certificates during the TLS handshake, so the IF-T Binding to TLS SASL client authentication might not be required unless TNC Server policy required an additional authentication of the TNC Client. Therefore, the normal usage for the SASL messages is when the TNC Client acted as the TLS client and did not authenticate during the TLS handshake.

4.7.1 SASL Authentication Requirements

Implementations compliant with the IF-T Binding to TLS specification MUST implement the SASL authentication messages described in this section. In order to ensure interoperability, all IF-T Binding to TLS version 2.0 implementations compliant with this specification MUST at least support the PLAIN SASL mechanism [PLAIN]. Similarly, implementations MUST provide the EXTERNAL SASL mechanism if both parties are authenticated during the TLS establishment. In order to be able to take advantage of other strong, widely deployed authentication technologies such as Kerberos and support for channel bindings, implementations MAY include support for GS2 (second GSS-API bridge for SASL) [GS2-MECH]. GS2 includes negotiable support for channel binding for use with SASL (see section 5 of RFC 5801).

4.7.2 SASL Use in IF-T Binding to TLS

SASL mechanism negotiation is initiated by the TNC Server sending the SASL Mechanisms message to the TNC Client to indicate the zero or more SASL mechanisms the TNC Server's policy is willing to use with the TNC Client. The TNC Client selects one SASL mechanism from the list and sends a SASL Mechanism Selection message completing the negotiation. Subsequent challenges and responses are carried within the SASL Authentication Data message carrying the authentication data for the selected mechanism. The authentication outcome is communicated in a SASL Result message containing a status code. If additional authentications are required, the TNC Server could trigger the next authentication by sending another SASL Mechanisms message after sending the SASL Result message for the current authentication mechanism.

4.7.3 SASL Authentication Flow

The SASL client authentication starts when the TNC Server enters the IF-T Negotiation phase and its policy indicates that an authentication of the TNC Client is necessary but was not performed during the TLS handshake protocol. The TNC Server is responsible for triggering the client authentication by sending the SASL Mechanisms message to the TNC Client listing the set of SASL mechanisms the server is willing to use based upon its policy.

The TNC Client selects a SASL mechanism from the list proposed by the TNC Server or sends an IETF Invalid Message error code in an IF-T Error message indicating it is unable or unwilling to perform any of the mechanisms that were offered. If the TNC Server receives a SASL Mechanism Selection message that contains an unacceptable SASL mechanism, the TNC Server would respond with an IETF PT-TLS SASL Mechanism Error Code in an IF-T Error message.

In situations where the TNC Server does not require a client authentication (either authentication isn’t necessary or was performed during the TLS Setup phase), the TNC Server MUST send a SASL Mechanisms message with no mechanisms included (only the IF-T header) indicating that the connection should transition to the IF-T Data Transport phase. The same mechanism is employed to indicate that a SASL authentication already performed in this session is adequate to permit transition to the IF-T Data Transport phase. So the TNC Server MUST always send a SASL Mechanisms message with no mechanisms as the last message in the IF-T Negotiation phase and the TNC Client MUST NOT transition to the IF-T Data Transport phase until it receives such a message.

If the TNC Server receives an IF-T assessment message before the completion of the client authentication, the TNC Server MUST send an IETF PT-TLS Authentication Required error code in an IF-T Binding to TLS Error message indicating to the TNC Client that an authentication exchange is required prior to entering the IF-T Data Transport phase.

4.7.4 Aborting SASL Authentication

The TNC Server may abort the authentication exchange by sending the SASL Result message with a status code of ABORT. The TNC Client may abort the authentication exchange by sending an IF-T Error message with an IETF namespace error code of SASL Mechanism Error.

4.7.5 Integration with SASL Framework

This sub-section discusses how the SASL framework integrates with the IF-T Binding to TLS protocol. The SASL “service name” for IF-T Binding to TLS is “nea-pt-tls”. This name is being used to be consistent with the PT-TLS protocol allowing them to be fully interoperable. SASL allows for authorization identity strings to be sent, but the IF-T Binding to TLS protocol does not make use of the SASL authorization identity string. SASL allows for mechanisms that support a “security layer” to protect the message exchange. The IF-T Binding to TLS operates under the protection of TLS so does not require SASL mechanisms to provide a security layer. Therefore, SASL security layers MUST be negotiated off during the SASL exchanges.

SASL is capable of supporting concurrent authentications, but the IF-T Binding to TLS only allows one SASL mechanism authentication to occur at one time. However, after a SASL authentication mechanism completes (successfully or unsuccessfully), the IF-T Binding to TLS allows the TNC Server to trigger an additional authentication by sending another SASL Mechanisms message.

4.7.6 SASL Channel Bindings

SASL channel bindings are used to bind the SASL authentication to the outer TLS tunnel to ensure that the authenticating endpoints are the same as the TLS endpoints. For SASL mechanisms that support channel bindings the TLS-unique value defined in RFC 5929 is carried by the SASL Mechanism. For most mechanisms this means including the tls-unique value with the appropriate prefix defined in RFC 5929 in the application data portion of the SASL Mechanism channel binding data. If the validation of the channel-binding fails then the connection MUST be aborted.

4.7.7 SASL Mechanisms Message

This message is sent by the TNC Server to indicate the list of SASL mechanisms that it is willing and able to use to authenticate the TNC Client.   Each mechanism name consists of a length followed by a name.   The total length of the list is determined by the message Length field.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Rsvd| Mech Len|             Mechanism Name (1-20 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Rsvd| Mech Len|             Mechanism Name (1-20 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Rsvd	This field MUST be set to 0 upon sending and MUST be ignored by compliant recipients.
Mech Len	This field contains the length of the Mechanism Name field in octets.
Mechanism Name	SASL mechanism name adhering to the rules defined in RFC4422.

4.7.8 SASL Mechanism Selection Message

This message is sent by the TNC Client in order to select a SASL mechanism for use on this session.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Rsvd| Mech Len|             Mechanism Name (1-20 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Optional Initial Mechanism Response              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
Rsvd	This field MUST be set to 0 upon sending and MUST be ignored by compliant recipients.
Mech Len	This field contains the length of the Mechanism Name field in octets.
Mechanism Name	SASL mechanism name adhering to the rules defined in RFC4422.
Optional Initial Mechanism Response	Initial set of authentication information required from the TNC Client to start the authentication. This data is optional and if not provided would be solicited by the TNC Server in the first SASL Authentication Data message request.

4.7.9 SASL Authentication Data Message

This message carries an opaque (to IF-T) blob of octets being exchanged between the TNC Client and the TNC Server. This message transports the SASL mechanism communications without interpreting any of the bytes. The SASL Authentication Data message MUST NOT be sent until a SASL mechanism has been established for a session. The SASL Authentication Data message associated with the current authentication mechanism MUST NOT be sent after a SASL Result is sent with a Successful status. Additional SASL Authentication Data messages would be sent if the IF-T Binding to TLS assessment initiator and responder desire a subsequent SASL authentication to occur but only after another SASL mechanism selection exchange occurs.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~            SASL Mechanism Data (Variable Length)              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description
SASL Mechanism Data	Opaque, variable length set of bytes exchanged between the IF-T assessment initiator's SASL mechanism and its peer IF-T assessment responder's SASL mechanism. These bytes MUST NOT be interpreted by the IF-T Binding to TLS layer.

4.7.10 SASL Result Message

This message is sent by the TNC Server at the conclusion of the SASL exchange to indicate the authentication result. Upon reception of a SASL Result message indicating an Abort, the TNC Client MUST terminate the current authentication conversation. The recipient may retry the authentication in the event of an authentication failure. Similarly, the TNC Server may request additional SASL authentication(s) be performed after the completion of a SASL mechanism by sending another SASL Mechanisms message including any mechanisms dictated by its 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Result Code         |     Optional Result Data      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            . . . . . . . . . . . . . . . . . . . . .          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Header Field	Description

<table>
<tr><td>Result Code</td><td>This field contains the result of the SASL authentication exchange.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Success

SASL authentication was successful and identity was confirmed.</td></tr>
<tr><td>1</td><td>Failure

SASL authentication failed. This might be caused by the client providing an invalid user identity and/or credential pair. Note that this is not a mechanism failure to process the authentication as reported by the Mechanism Failure code.</td></tr>
<tr><td>2</td><td>Abort

SASL authentication exchange was aborted by the sender.</td></tr>
<tr><td>3</td><td>Mechanism Failure

SASL “mechanism failure” during the processing of the client’s authentication (e.g. not related to the user’s input).</td></tr>
</table></td></tr>
<tr><td>Optional Result Data</td><td>This field contains a variable length set of additional data for a successful result. This field MUST be zero length unless the TNC Server is returning a Result Code of Success and has more data to return.   For more information on the additional data with success in SASL, see RFC 4422.</td></tr>
</table>

4.8 IF-T Error Message

This section describes the format and contents of the IF-T (also PT-TLS) Error Message sent by the NAR or NAA when it detects an IF-T level protocol error. Each error message contains an error code indicating the error that occurred, followed by a copy of the message that caused the error.

When an IF-T error is received, the recipient MUST NOT respond with an IF-T error because this could result in an infinite loop of error messages being sent. Instead, the recipient MAY log the error, modify its behavior to avoid future 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 Message Value portion of an IF-T Error Message contains the following information:

```
                     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                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Copy of Original Message (Variable Length)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           . . . . . . .                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

<table>
<tr><th>Header Field</th><th>Description</th></tr>
<tr><td>Reserved</td><td>Reserved for future use. This field MUST be set to zero on transmission and ignored upon reception.</td></tr>
<tr><td>Error Code Vendor ID</td><td>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 other vendor-defined Error Code name spaces this field MUST be set to the SMI Private Enterprise Number of the vendor. Version 2 of the IF-T Binding to TLS leverages the IETF PT-TLS set of error codes to enable interoperability. The IETF uses a value of 0 in this field.</td></tr>
<tr><td>Error Code</td><td>This field contains the error code. This error code exists within the scope of the Error Code Vendor ID in this message. The NAA and NAR MUST NOT require support for particular vendor-specific IF-T Error Codes and MUST interoperate with other parties despite any differences in the set of vendor-specific IF-T Error Codes supported (although they MAY permit administrators to configure them to require support for specific IF-T error codes).

Version 1.0 of the IF-T Binding to TLS defined a set of Error Code values in the TCG name space. However in order to establish interoperability with IETF PT-TLS implementations, Version 2.0 of this specification leverages the IETF defined Error Codes and does not define any new Error Codes for the TCG name space.

When the Error Code Vendor ID is set to the IETF Private Enterprise Number, the following table lists the supported standard numeric error codes:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Reserved

Reserved value indicates that the IF-T Error Message SHOULD be ignored by all recipients. This MAY be used for debugging purposes to allow a sender to see a copy of the message that was received while a receiver is operating on its contents.</td></tr>
<tr><td>1</td><td>Malformed Message

IF-T message unrecognized or unsupported. This error code SHOULD be sent when the basic sanity test fails when checking the IF-T message header. The sender of this error code MUST consider it a fatal error and abort the assessment.</td></tr>
</table>
</td></tr>
</table>

2	Version Not Supported This error SHOULD be sent when an assessment responder receives an IF-T Version Request message containing a range of version numbers outside the range the recipient is willing and able to support on the session. All IF-T Binding to TLS messages carrying the Version Not Supported error code MUST use a Version number of 1. All parties that receive or send IF-T Binding to TLS messages MUST be able to properly process an error message that meets this description, even if they cannot process any other aspect of IF-T version 1. The sender of this error code MUST consider it a fatal error and close the TLS session after sending this IF-T message.
3	Type Not Supported IF-T message type unknown or not supported. When a recipient receives a IF-T message type that it does not support, it MUST send back this error, ignore the message and proceed. For example, this could occur if the sender used a Vendor ID for the Message Type that is not supported by the recipient. This error message does not indicate a fatal error has occurred, so the assessment is allowed to continue.
4	Failed Authentication The authentication of the identity of the client failed. This could occur if the SASL mechanism was unable to authenticate the claimed identity of the TNC Client. This error message does not indicate a fatal error has occurred, so the authentication is allowed to be re-started.
5	Invalid Message IF-T Binding to TLS message received was invalid based on the protocol state. For example, this error would be sent if a recipient receives a message associated with the IF-T Negotiation Phase (such as Version messages) after the protocol has reached the IF-T Data Transport Phase. The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this IF-T message.
6	SASL Mechanism Error A fatal error occurred while trying to perform the client authentication. For example, the TNC Client is unable to support any of the offered SASL mechanisms. The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this IF-T message.
7	Invalid Parameter

	The IF-T Binding to TLS error message sender has received a message with an invalid or unsupported value in the IF-T header. This could occur if the TNC Client receives a IF-T message from the TNC Server with a Message Length of zero. The sender and receiver of this error code MUST consider it a fatal error and close the TLS session after sending or receiving this IF-T message.
Copy of Original Message	This variable length value MUST contain a copy (up to 1024 bytes) of the original IF-T message that caused the error. If the original message is longer than 1024 bytes, only the initial 1024 bytes will be included in this field. This field is included so the error recipient can determine which message sent caused the error. In particular, the recipient can use the Message Identifier field from the Copy of Original Message to determine which message caused the error.

5 Security Considerations

This section discusses the countermeasures provided by the IF-T Binding to TLS protocol to the threats that each binding of IF-T transport protocol must face. Rather than replicate much of the security considerations from the IF-T Binding for Tunneled EAP Methods [IF-T-EAP], readers are directed to read section 5 of that specification to understand the threat environment and minimum security protections expected from IF-T.

Section 5.4.2 of the IF-T Binding for Tunneled EAP Methods establishes some common requirements that are to be addressed by all IF-T bindings. These 5 requirements are:

1. Cryptographic authentication of the NAA to the NAR
2. NAR authentication and TNC dialog protected by at least a cryptographic transport
3. Encryption of the message stream tied to at least the transport authentication
4. Cryptographic integrity protection of the message tied to at least the transport authentication
5. Protection against replay attack

The TLS binding of IF-T meets each of these requirements leveraging the cryptographic protections inherent in TLS. The following list discusses how each corresponding requirement is met by the protections provided by TLS:

1. All implementations of the IF-T Binding to TLS MUST support at least server (NAA) side certificate based authentication to the NAR using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite. Deployers may choose to use other methods of authentication of the server, but all implementations will offer at least support for server certificate based authentication. When a TNC Client is capable of operating as the TLS server (accepting inbound IF-T TCP connections from the TNC Server), the TNC Client MUST also support the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite and have a certificate.
2. All implementations compliant with this specification MUST enable the NAR to authenticate using the PLAIN or the EXTERNAL SASL mechanism inside the cryptographically protected TLS record layer although they MAY allow this capability to be disabled by configuration. Implementations SHOULD also support cryptographic authentication mechanisms (e.g. client side TLS certificates) to allow the NAR to authenticate during the TLS handshake prior to the protected TLS record layer. The TNC dialog (IF-TNCCS messages) is carried over the TLS record layer after a suitable cryptographic transport has been established. This allows for the entire TNC assessment to be protected from eavesdropping while on the network. Implementations of this specification targeting use on TCG trusted platforms SHOULD provide support for using client side TLS certificates containing the SKAE extension and allow for the use of the certificate with the trusted platform's trusted roots during attestation (e.g. signing an attestation Integrity Report using the SKAE extended certificate).
3. Because the TLS handshake protocol must support at least NAA side certificates, the resulting TLS record layer carrying the IF-T message stream will be protected by keys associated with the NAA authentication.
4. Again, the TLS handshake protocol will be capable of using a cryptographic authentication of the NAA using certificates. While under the protection of the NAA authentication exchange, keys are established to protect the integrity of each IF-T message sent using the TLS record layer.
5. There are a number of potential types of replay attack. Passive eavesdropping with later replay of observed information attacks are thwarted by the secrecy protection offered by the encrypting of the TLS record layer carrying the IF-T messages. Active replay attacks are addressed by the strong cryptographic authentication of the NAA by the NAR, thus

preventing rogue (untrusted) third parties from becoming a man in the middle intercepting and relaying messages. Similarly, the TLS record layer protects the NAR authentication from eavesdropping and replay.

Finally, section 5.4.5 of the IF-T Binding for Tunneled EAP Methods describes an active man in the middle attack where an adversary controlling a trusted NAA could trick a clean endpoint to provide compliant TPM based measurements. These measurements are recorded and replayed during an access-time assessment with a target network in order to appear compliant to gain access. Several possible countermeasures exist when the TNC client is integrated with TPM. The IF-T Binding to TLS specifies the use of tls-unique to establish a shared secret between the TNC Client and Server which is provided to the PTS-IMC and PTS-IMV for integration into a TPM-based attestation. IF-T Binding to TLS implementations MUST support the tls-unique capability when executing on a platform with an enabled TPM to provide a countermeasure to the Asokan attack.

Another possible countermeasure to the Asokan attack involves the use of the SKAE extension to the TNC Client's certificate used during the authentication portion of the TLS handshake. The SKAE extension provides evidence that the private key associated with the public key contained in the certificate is housed and protected inside of the platform's TPM. The use of the certificate during the TLS handshake ensures that the TNC Client's identity and TPM resident private key are incorporated into the establishment of the TLS session keys. After the IF-T session has been established over TLS, when an assessment occurs the TNC Server SHOULD make use of an IMV supporting a TPM-based attestation (e.g. using the PTS). This assessment leverages the TPM resident key to offer a signature over the assessment data and quote linking the reported measurements to the key known to be present on the TLS authenticated endpoint. These protections parallel those offered in the IF-T Binding for Tunneled EAP Methods and version 1.0 of this specification, which also is able to leverage SKAE extensions to X.509 certificates.

5.1 Trust Relationships and Countermeasures

Version 2.0 of the IF-T Binding to TLS is a fully interoperable protocol with the IETF PT-TLS protocol 1.0. Therefore, readers interested in a better understanding of the trust model associated with the primary IF-T components and potential classes of attacks should read the Security Considerations section of the PT-TLS specification.

6 Privacy Considerations

The role of IF-T is to act as a secure transport for IF-TNCCS and other higher layer protocols. As such, IF-T does not directly utilize personally identifiable information (PII) except when client authentication is enabled. When client authentication is being used, the TNC Client may be asked to disclose a local identifier (e.g. username) associated with the endpoint and an authenticator (e.g. password) to authenticate that identity. Because the identity and authenticator are potentially privacy sensitive information, the TNC Client MUST include a mechanism to restrict which TNC Servers will be sent this information. Similarly the TNC Client SHOULD provide an indication to the person being identified that a request for their identity has been made in case they choose to opt out of the authentication in order to remain anonymous unless no user interface is available. Whether a TNC Client must obtain permission to reveal a person's identity depends on whether permission has already been granted, and is subject to local law and regulations.

IF-T provides cryptographic peer authentication, message integrity, and data secrecy to higher layer TNC protocols that may exchange data potentially including PII. These security services can be used to protect any PII involved in an assessment from passive and active attackers on the network. Endpoints sending potentially privacy sensitive information SHOULD ensure that the IF-T security protections (TLS cipher suites) negotiated for a TNC assessment of the endpoint are adequate to avoid interception and off-line attacks of any long term privacy sensitive information unless other network protections are already present.

7 References

7.1 Normative References

[BINDINGS] Altman, J., Williams, N., Zhu L., “Channel Bindings for TLS”, RFC 5929, July 2010.

[CRL] Cooper, D., Santesson, S., Farrel, S., Boeyen, S., Housley, R., Polk, W., “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, RFC 5280, May 2008.

[HEARTBEAT] Segglemann, R., Tuexen, M., Williams M., “Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension”, RFC 6520, February 2012.

[INTREPORT] Trusted Computing Group, “TCG Infrastructure Integrity Report Schema Specification”, https://www.trustedcomputinggroup.org/specs/IWG/IntegrityReport_Schema_Specification_v1.0.pdf, November 2006.

[IF-PTS] Trusted Computing Group, “TCG Infrastructure Platform Trust Services Specification (IF-PTS)”, https://www.trustedcomputinggroup.org/specs/IWG/IF-PTS_v1.0.pdf, November 2006.

[KEYWORDS] Bradner S., "Keywords for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[NAME-VALID] Saint-Andre, P., Hodges, J., “Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of

Transport Layer Security (TLS)”, RFC 6125, March 2011.

[PLAIN] Zeilenga K., “The PLAIN Simple Authentication and Security Layer (SASL) Mechanism”, RFC 4616, August 2006.

[PT-TLS] Sangster P., Cam-Winget N., Salowey J., "PT-TLS: A TCP-based Posture Transport (PT) Protocol", RFC 6876, February 2013.

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

[SKAE] Trusted Computing Group, “TCG Infrastructure Subject Key Attestation Evidence Extension”, http://www.trustedcomputinggroup.org/specs/IWG/IWG_SKAE_Extension_1-00.pdf, June 2005.

[TLS11] T. Dierks, E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.1”, RFC 4346, April 2006.

[TLS12] T. Dierks, E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, RFC 5246, August 2008.

[TLS-RENEGO] Rescorla E., Ray M., Oskov N., “Transport Layer Security (TLS) Renegotiation Indication Extension”, RFC 5746, February 2010.

[UTF8] F. Yergeau, “UTF-8, a transformation format of ISO 10646”, RFC 3629, November 2003.

7.2 Informative References

[ASOKAN] Salowey, J., Hanna, S., “The Network Endpoint Assessment (NEA) Asokan

Attack Analysis”, RFC 6813, December 2012.

[BINDING-TLS] Altman, J., Williams, N., Zhu L., “Channel Bindings for TLS”, RFC 5929, July 2010.

[EAP] Aboba B., “Extensible Authentication Protocol”, RFC 3748, June 2004.

[GS2-MECH] Josefsson, S., Williams, N., "Using Generic Security Service Application Program Interface (GSS-API) Mechanisms in Simple Authentication and Security Layer (SASL): The GS2 Mechanism Family", RFC 5801, July 2010.

[TNC-ARCH] Trusted Computing Group, “TNC Architecture for Interoperability”, https://www.trustedcomputinggroup.org/specs/TNC/TNC_Architecture_v1_5_r3.pdf, May 2012.

[IF-T-EAP] Trusted Computing Group, “TNC IF-T: Protocol bindings for Tunneled EAP Methods”, https://www.trustedcomputinggroup.org/specs/TNC/TNC_IFT_v1_1_r10.pdf, May 2007.

[IF-IMC] Trusted Computing Group, “TNC IF-IMC”, https://www.trustedcomputinggroup.org/specs/TNC/TNC_IFIMC_v1_3_r18.pdf, February 2013.

[IF-IMV] Trusted Computing Group, “TNC IF-IMV”, https://www.trustedcomputinggroup.org/specs/TNC/TNC_IFIMV_v1_3_r12.pdf, February 2013.

[IF-T-TLS1] Trusted Computing Group, "TNC IF-T: Binding to TLS", http://www.trustedcomputinggroup.org/files/resource_files/51F0757E-1D09-

3519-AD63B6FD099658A6/TNC_IFT_TLS_v1_0_r16.pdf, May 2009

[IF-TNCCS12] Trusted Computing Group, “TNC IF-TNCCS”, http://www.trustedcomputinggroup.org/files/resource_files/51ED9FF7-1D09-3519-AD3E3B2EBEECEB3F/TNC_IF-TNCCS_v1_2_r6.pdf, May 2009.

[IF-TNCCS20] Trusted Computing Group, “TNC IF-TNCCS”, http://www.trustedcomputinggroup.org/files/resource_files/495CA3DD-1D09-3519-AD0043966E821ECB/IF-TNCCS_TLVBinding_v2_0_r16a.pdf, January 2010.

[IF-TNCCS-SOH] Trusted Computing Group, “TNC IF-TNCCS: Protocol Bindings for SoH”, https://www.trustedcomputinggroup.org/specs/TNC/IF-TNCCS-SOH_v1.0_r8.pdf, May 2007.