TCG Trusted Network Communications TNC Architecture for Interoperability

Specification Version 2.0
Revision 13
16 October 2017
Published

Contact:
admin@trustedcomputinggroup.org

TCG

TCG PUBLISHED

Disclaimer

Acknowledgement

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

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

Scott Kelly	**Aruba Networks**
David Louin	**AMOSSYS**
Maxime Olivier	**AMOSSYS**
Adrien Raffin	**AMOSSYS**
Amit Agarwal	**Avaya**
Mahalingam Mani	**Avaya**
Jeffery Dion	**Boeing**
Greg Kimberly	**Boeing**
Steven Venema	**Boeing**
Phil Dollery	**CESG**
Peter Wrobel	**CESG**
Nancy Cam-Winget	**Cisco Systems**
Max Pritikin	**Cisco Systems**
Mark Townsend	**Enterasys**
Michael McDaniels	**Extreme Networks**
Henk Birkholz	**Fraunhofer SIT**
Andreas Fuchs	**Fraunhofer SIT**
Hidenobu Ito	**Fujitsu Limited**
Seigo Kotani	**Fujitsu Limited**
Houcheng Lee	**Fujitsu Limited**
Sung Lee	**Fujitsu Limited**
Gerald Maunier	**Gemalto**
Graeme Proudler	**Hewlett-Packard**
Mauricio Sanchez	**Hewlett-Packard**
Ira McDonald	**High North**
Andreas Steffen	**HSR**
Ren Lanfang	**Huawei Technologies**
Jiwei Wei	**Huawei Technologies**
Shi Xun	**Huawei Technologies**
Han Yin	**Huawei Technologies**
Yi Zhang	**Huawei Technologies**
Diana Arroyo	**IBM**
Guha Prasad Venkataraman	**IBM**
Sean Convery	**Identity Engines**
Chris Hessing	**Identity Engines**
Morteza Ansari	**Infoblox**
Stuart Bailey	**Infoblox**
Ivan Pulleyn	**Infoblox**
Ravi Sahita	**Intel Corporation**
Ned Smith	**Intel Corporation**
Josh Howlett	**JANET (UK)**
Yan Avlasov	**Juniper Networks**
Roger Chickering	**Juniper Networks**

Charles Goldberg	**Juniper Networks**
Steve Hanna (Editor)	**Juniper Networks**
PJ Kirner	**Juniper Networks**
John Jerrim	**Lancope**
Tom Price	**Lumeta**
Matt Webster	**Lumeta**
Ryan Hurst	**Microsoft**
Atul Shah (TNC co-chair)	**Microsoft**
Charles Schmidt	**MITRE**
Sandilya Garimella	**Motorola**
Meenakshi Kaushik	**Nortel**
Cliff Kahn	**Pulse Secure**
Lisa Lorenzin (Editor, TNC co-chair)	**Pulse Secure**
Carolin Latze	**Siroop**
Arrive Fabien	**ST Microelectronics**
Christopher Ernst	**Swisscom**
Thierry Hayoz	**Swisscom**
Paul Sangster	**Symantec Corporation**
Anne-Rose Gratadour	**Thales**
Brad Upson	**UNH InterOperability Lab**
Mike Boyle	**US Government**
Jessica Fitzgerald-McKay	**US Government**
Lauren Giroux	**US Government**
Chris Salter	**US Government**
Dave Waltermire	**US Government**
Thomas Hardjono	**Wave Systems**
Greg Kazmierczak	**Wave Systems**

Table of Contents

1 Scope and Audience

1.1 Scope

TCG's Trusted Network Communications (TNC) network security architecture and open standards enable intelligent policy decisions, dynamic security enforcement, and communication between security systems. TNC standards provide network and endpoint visibility, helping network managers know who and what is on their network, and whether devices are compliant and secure. TNC standards also enable context-based access control enforcement - granting or blocking access based on authentication, device compliance, and user behavior - and security automation, for orchestration of network and security systems. The current list of TNC specifications is available on the TCG website at http://www.trustedcomputinggroup.org/work-groups/trusted-network-communications/open-standards-tnc/.

TNC addresses four main classes of problems. ***Network visibility*** asks who is on the network, and what they are trying to access. ***Endpoint compliance*** asks whether devices on the network are secure, and whether user/device behavior is appropriate. ***Network enforcement*** requires the ability to block unauthorized users, devices, and/or behaviors, and to grant appropriate levels of access to authorized devices. ***Security automation*** requires sharing real time information about the environment without giving out sensitive, private, or protected data; and asks how to benefit from threat intelligence generated across the spectrum, both internally to the environment and externally from other sources.

The TNC architecture offers three primary capabilities:

- a ***Compliance*** capability, which evaluates an endpoint's adherence to network policy both at the point of connection and while it is connected to the network;
- an ***Orchestration*** capability, which provides a dynamic repository and notification service for real-time state and events; and
- an ***Access Control*** capability, which controls access to protected resources and networks based on endpoint posture and many other factors.

These cross-domain capabilities can be used for many purposes, including - but not limited to - security automation, continuous monitoring, asset management, endpoint compliance assessment and enforcement, protection of critical resources, leveraging of shared information, event correlation and assessment, and a variety of other key buzzword-compliant[1] functions. Application of these capabilities enables trusted network communications - the ability to understand the trustworthiness of an endpoint before, and while, it's allowed to communicate on the network.

1.2 Audience

This document is written with two broad classes of users in mind: users of the document itself, whose goal may be to develop products, design networks, and/or draft other standards based on the TNC Architecture, and users of the technology enabled by the TNC Architecture. In other words:

- product implementers, solution architects, specification and standards developers, and others who are interested in the development, deployment, and interoperation of environments with trusted network communications and reliable, resilient endpoints may find this document helpful in understanding the TNC architecture and specifications; and

[1] In all seriousness, these concepts are buzzwords for a reason; they represent critical functions that span operational use cases and disparate environments.

- solution architects, solution implementers, and systems administrators interested in designing secure networks may find this document helpful in understanding how TNC-enabled technologies can facilitate that goal.

A more extensive overview of intended audience and envisioned usage is provided in Appendix A: User Communities.

The TNC Architecture specifies how the body of TNC specifications relate to each other and to the use cases and capabilities they enable. Accordingly, this document does not provide normative requirements; normative requirements for each TNC interface or profile are contained in the specification for that interface or profile.

2 Introduction

2.1 Background

The constant innovation that makes technology so appealing - and so essential - also makes digital security a moving target. The explosive growth of mobile devices, our increasing reliance on wireless networks, and the Internet of Things - the interconnection of physical objects through the Internet that promises more personalized experiences and deeper integration of technology in our lives - brings with it an ongoing wave of new security threats which require new and innovative ways of dealing with them.

Data breaches, cybersecurity attacks leading to loss of intellectual property and security, are serious threats to personal privacy, business integrity and national security worldwide. Yet despite ongoing efforts aimed at eliminating the security vulnerabilities that lead to such breaches and attacks, they are becoming increasingly pervasive and complex.

The threat environment has changed significantly, with increases in unmanaged or less managed devices (e.g., consumerization of IT), unknown software in the environment, and hostile networks driving demand for always-on finer-grained security controls and real time monitoring and protection of resources and data.

IT security standards play an ever-increasing role in striking a balance between consistent personalized experiences on a variety of devices and environments on the one hand, and the need to contain cost and minimize barriers to commerce on the other. IT security standards need to innovate and evolve in order to address:

- Risk management across a wide range of computing devices in cost effective efficient ways
- Growing risks of endpoint compromise and data loss in an ever-increasing connected cyber world
- Increased need to share security information and threat protections with others
- Automated cyber-attacks that necessitate the need for automated cyber defenses

2.1.1 Trusted Network Communications

The Trusted Computing Group (TCG) is an international not-for-profit standards organization that develops, defines, and promotes open, vendor-neutral specifications for interoperable trusted computing platforms. TCG participants include business and technical specialists from the world's leading silicon makers, device manufacturers, and software and solution providers. Working collaboratively with industry experts, government officials, and academic researchers, TCG has been advancing trusted computing technology worldwide for more than a decade.

In response to the global need for a more secure computing environment, TCG has developed and published Trusted Network Communications (TNC) standards since 2005, as an open architecture originally intended as a network access control standard with a goal of multi-vendor endpoint policy enforcement. In 2009, TCG announced expanded specifications which extended the scope of TNC to include security automation. Additional real-world applications of TNC include Industrial Control System (ICS) & SCADA security, as well as endpoint compliance and continuous monitoring. The TNC Architecture continues to evolve, expanding the existing end-to-end trust fabric from traditional use cases to emerging areas such as network infrastructure, Internet of Things (IoT), mobility, and cloud applications. TNC standards integrate security components across the endpoint, network, and servers into an intelligent, responsive, coordinated defense.

The use of standardized protocols and schema offer benefits to both users and implementers of technology solutions. For users, the use of publicly-vetted protocols helps secure data in transit, reduces dependency on single-vendor solutions, and allows for architectural flexibility that meets

the needs of myriad use cases. For implementers, standards can help meet the demands of international and national regulatory bodies, simplify interoperability with other vendor products, and provide more value to customers.

2.1.2 TNC Elements

TNC elements are the fundamental building blocks of the architecture:

Element	Definition
Function	An element providing a unitary operation of the TNC Architecture
Component	An element (usually a piece of software) implementing one or more functions of the TNC Architecture
Interface	A standardized method of communication between components
Role	An actor with a specific purpose in the TNC Architecture, utilizing a particular set of functions
Entity	A device (physical or logical) performing one or more roles in the TNC Architecture

2.1.3 Introducing the TNC Architecture

The TNC Architecture recognizes the following high-level roles for entities involved in trusted network communication:

- ***Enforcement points***, which consume access control decisions from a policy server and apply them to endpoint requests
- ***Policy servers***, which collect and evaluate endpoint posture information and/or make access control decisions based on endpoint context (including role, state, location, behavior, and other factors) and communicate those decisions to enforcement points
- ***Configuration Management Databases (CMDBs)***, which store collected endpoint measurements
- ***CMDB clients***, which communicate endpoint information to and consume it from CMDBs,
- ***Metadata Access Points (MAPs)***, which provide centralized coordination for producers and consumers of network and security information
- ***MAP clients***, which publish, search for, and subscribe to updates on endpoint and environment information via a MAP

Endpoints are also an important component of the TNC Architecture. Unfortunately, the term "endpoint" is horribly overloaded. In this document, the following conventions are used:

- The term "endpoint" (lower-case "e") refers to any entity - physical or virtual - that can be connected to a network (including infrastructure devices such as routers and servers, as well as end-user devices such as workstations and mobile devices). This corresponds to the IETF definition of endpoint in RFC 5209 [1].
- The term "Endpoint" (capital "E") specifically refers to an endpoint interacting with a Policy Server and performing TNC Client-related functions as defined in section 5.3.1.

A single entity may take on multiple roles; for example, a policy server may also be a MAP client as well as a CMDB client. See Section 5.2 for more detail on these roles. Figure 1 illustrates the relationship between TNC roles:

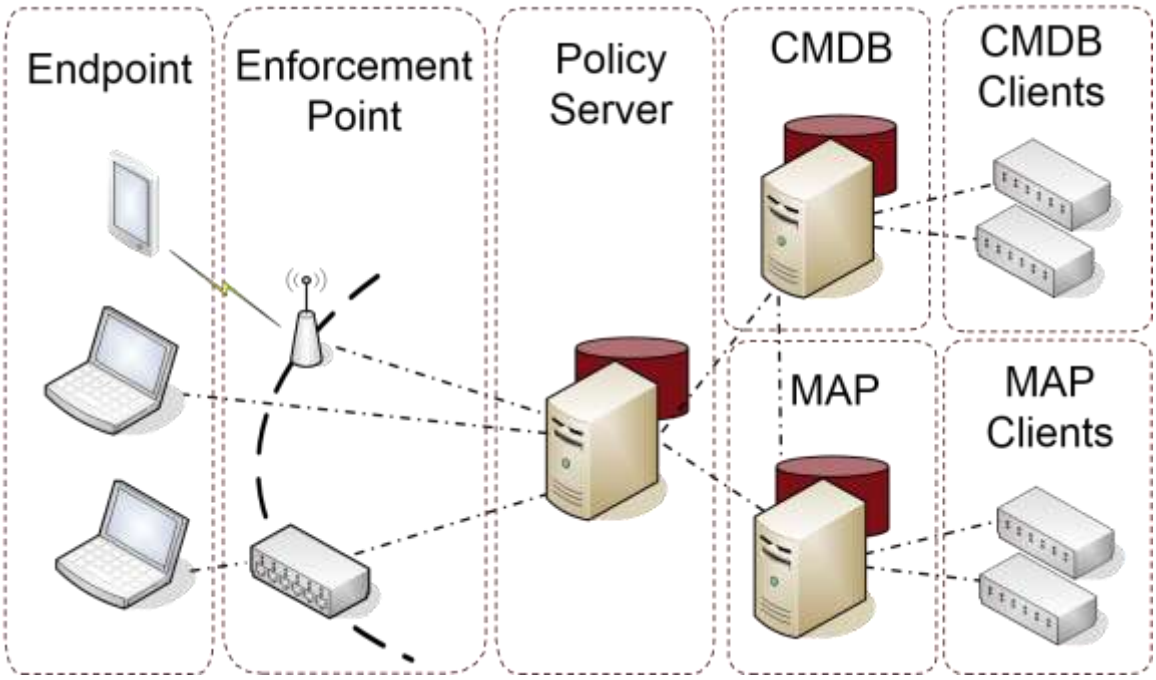

Figure 1: TNC Architecture Overview

2.1.4 Measuring Endpoint State

One of the primary purposes of the TNC standards is to enable the collection and evaluation of endpoint state information, enabling assessment of endpoint and network health. Endpoint *posture* is the aggregate state of the endpoint [1], composed by collecting measurements of specific aspects of endpoint state. TNC originally focused on *integrity measurements,* primarily around integrity of the BIOS / operating system and security controls (such as current anti-virus signature file or running endpoint firewall); TNC measurements have since expanded to include broader *posture measurements*, such as endpoint provisioning, installed software, and/or configuration. All of these measurements are collected via an *integrity check handshake*, an exchange of information between TNC components. Endpoint *compliance* indicates whether the endpoint posture meets the requirements of its environment.

2.2 Relationships to Other Standards

2.2.1 Relationship to TPM

The TCG's Trusted Platform Module (TPM) [2] [3] supports hardware-based "roots of trust", which help establish platform integrity, user security, and privacy. The TPM enables a “chain of trust” for a device’s core components responsible for its boot process and ultimately the execution of the OS and applications.

In the context of TNC architecture and solutions, TNC can leverage a TPM to increase trust in endpoint measurements, which improves detection of compromised devices so that appropriate controls can be applied. TNC mechanisms enable expression of that trust to a third party or back-end verifier, increasing confidence both in endpoint evaluation and in resulting actions such as access control decisions. See section 8 for a discussion of TNC with TPM.

2.2.2 Relationship to TCG IWG Architecture

The TNC Architecture builds upon the original architecture developed by the TCG's Infrastructure Working Group (IWG), which outlines a Platform Authentication model [4]. See Appendix B: Relation to TCG IWG Architecture for discussion of the TNC Architecture in relation to the IWG

Architecture.

2.2.3 Relationship with IETF NEA

Several TNC endpoint posture-related specifications have been adopted by IETF as the basis for the IETF Network Endpoint Assessment standards. The IETF NEA Posture Assessment [5], Posture Broker [6], and Posture Transport [7] [8] protocols are all based on TNC interfaces. See Section 7.1.1 for a more detailed mapping between TNC and NEA terms and concepts.

2.3 Aim and Purposes

TCG's Trusted Network Communications (TNC) network security architecture and open standards enable intelligent policy decisions, dynamic security enforcement, and communication between security systems. TNC standards facilitate network and endpoint visibility, helping network managers know who and what is on their network and whether devices are compliant and secure. TNC standards also enable network-based access control enforcement - granting or blocking access based on authentication, device compliance, and user behavior - and security system integration - real-time information sharing enabling dynamic integration of network and security products.

The aim of the TNC architecture is to provide a framework for the development of standards to support multi-vendor solutions for:

- *Endpoint compliance:* TNC specifies schema and protocols for standardized endpoint posture reporting, both self-reporting and by observing parties. TNC-enabled technology provides systems administrators the ability to know the aggregate endpoint compliance status of their environment with near-real-time updates, and reduce the fragmentation of endpoint posture reports in proprietary databases. This allows the data to be shared across the network, with network and administrative tools and with other TNC Architecture elements, to support asset management, threat detection, security automation and vulnerability analysis use cases.

  The TNC Architecture facilitates remediation of endpoints which fail posture verification by assisting in detection of endpoints requiring remediation and providing a transport mechanism for remediation instructions; however, it does not standardize specific methods of remediation. See Appendix C: Assessment, Isolation, and Remediation for details.

- *Control of access to resources:* TNC-enabled technology can serve as a gatekeeper for endpoints accessing sensitive resources, including but not limited to access to a given network, particular services on the network, specific applications, and/or information. TNC enables this control either by dynamic configuration of devices (such as switches and firewalls) that are specifically intended to allow or deny access to resources based on policy decisions, or by making information available to specific services, which then used this provided information to make their own access control decisions. The result is that TNC facilitates a coordinated, multi-level, and comprehensive approach to ensuring that sensitive resources and actions are only available to authorized parties. See Appendix D: Basic Message Flows for Network Admission for details.

- *Policy enforcement:* As noted above, TNC can help prevent unauthorized endpoints and users from accessing sensitive information and resources. TNC gives administrators a great deal of power to defined what "unauthorized" means in each case. Decisions as to whether a given request is authorized can hinge on many factors, including but not limited to user identity, device identity, the degree to which the endpoint is considered compliant with policy, time of day, location of the accessing device, type of device (e.g., laptop, mobile device, etc.), and other contextual information that sensors and other parties may have reported about the requesting device and its activities. The criteria for authorization can be as simple or sophisticated as necessary. Since TNC allows access

to such a broad array of information, the sources of information are extensible, and all such information can be given weight in making an authorization decision, administrators benefit from the ability to carefully tune when, how, why, and by whom a given action is performed.

- *Security automation:* Security automation is the ability to automate and streamline known default tasks or responses, so they can be repeated quickly and as needed. TNC architecture and elements provide robust means of detection of and response to security events across the enterprise. TNC-enabled technology can share information in real time, using standard protocols and extensible data formats, which helps all security systems to be more intelligent by giving them the information they need to get their job done better and faster. TNC enables security automation protections needed to help defend resources and networks against today's security threats as well as the evolving security landscape in the future.
- *Threat detection and response*: In addition to the ability to limit access for unauthorized endpoints, TNC enables continuous monitoring of endpoint posture and behavior, which gives administrators an expanded ability to find and remediate vulnerabilities. TNC standards facilitate measurement of how effectively network policy was implemented, enable dynamic, intelligent response to non-compliant endpoint state and/or behavior, and allow the enterprise policy to be responsive to external factors such as new threat intelligence, new vulnerability reports, etc.

2.4 Benefits of TNC

TNC standards deliver a wide range of benefits:

- TNC standards have a proven track record of delivering interoperable solutions to address endpoint, network, and server security. Products based on TNC standards have been shipping since 2005. There are many open-source implementations as well.
- TNC standards are widely deployed in real production scenarios. A broad range of customers across many sectors (Government, Healthcare, Finance, Retail and Education, among others) are benefitting from interoperable security solutions based on TNC standards.
- TNC standards are completely vendor-neutral. TNC based solutions leverage existing network infrastructure in a production environment, adding value to the existing investment.
- TNC standards are flexible. They support a broad range of assessment options (identity, health, behavior, and location; hardware-based & software-based security; and pre-admission & post-admission evaluation and monitoring). TNC standards also accommodate rapid change and can adapt to the evolving security landscape.
- TNC standards can and do easily integrate with other standards both existing and emerging, e.g. SWID Tags (ISO 19770-2) [9].

3 Capabilities Enabled by the TNC Architecture

The TNC Architecture is composed of a set of interfaces between components that enable various capabilities. Currently, TNC is focusing on three fundamental capabilities:

1. Compliance capability - evaluates an endpoint's adherence to network policy while it is connected to the network
2. Orchestration capability - provides a dynamic repository and notification service for real-time state and events
3. Access Control capability - controls access to protected resources and networks based on endpoint posture and many other factors.

Each of these capabilities can operate independently, which allows organizations to implement the features their network security plan requires. These capabilities can also act in concert, which provides abundant data for analyzing the state of enterprise network security. These are the three foundational capabilities for permitting authorized access and preventing, detecting, and responding to unauthorized access and network attacks.

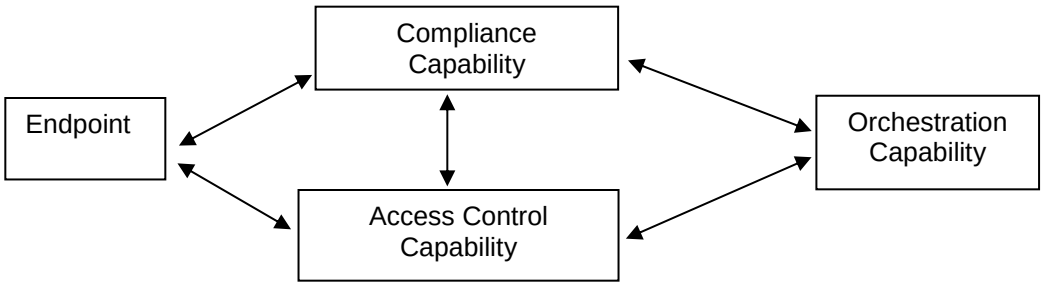

Figure 2: TNC Architecture Capabilities

The TNC defines an endpoint as any network-connected device. Therefore, the architecture is designed to enable posture checking, behavior monitoring, and remediation not only of user endpoints, such as PCs, laptops, phone, and other mobile devices, but also of infrastructure devices that are continually connected to the network and that are highly valuable targets of attack. The TNC architecture allows administrators to answer the questions

- "Is it vulnerable?"
- "Is it compromised?"
- "What actions should be permitted?"

for all the endpoints on their network and, more broadly, for the network itself.

This outline describes a set of capabilities that can be implemented using TNC protocols and interfaces. An implementer may choose to design a single capability on one server, or combine capabilities on a server. Either approach is correct, and network architects can choose the approach that best serves their needs. Furthermore, these are not the only capabilities capable of being supported by the architecture. TNC protocols and interfaces can be combined in myriad ways to solve many different problems. This outline details three examples.

3.1 The TNC Compliance Capability

The Compliance capability enables collection and evaluation of endpoint compliance information, consuming information from the endpoint and/or from the Orchestration capability about the endpoint. Values received, either from the compliance report or from the results of a scan, can be compared to an environment's compliance policy.

The Compliance capability enables an administrator to collect compliance reports from endpoints and evaluate these reports against network policy to identify non-compliant endpoints; the reports may be stored for future reference by authorized administrators and network tools (such as asset

management, threat defense, and reporting tools). It also allows an administrator to send queries to see what applications an endpoint reported as having installed, and can trigger an updated report from the endpoint.

The administrator can also run scans of the endpoint to check for signs of intrusion or to check low-level configuration variable. Information gathered during these scans can be compared against network policy, and used to make remediation decisions, including decisions to update, quarantine or remove an endpoint from the network. Scan results can also be stored for future reference.

3.2 The TNC Orchestration Capability

The Orchestration capability offers a notification service and unified, extensible data model. Network and security devices can benefit from context they obtain from the Orchestration capability to better perform their functions, and can share context with the Orchestration capability to enable other elements to better perform their functions. For example, a Security Information and Event Management system (SIEM) can share information about activity on the network, enabling a policy server to make more informed decisions about an endpoint; the policy server can share information about the characteristics or state of an endpoint, enabling the SIEM to apply device-specific analytics.

Participating network and security devices can subscribe to information and publish information via the Orchestration capability. This allows for robust information sharing across the network. Examples of endpoints that might benefit from the information passing through the Orchestration capability include sensors that monitor and report on observed network behavior; network tools (such as policy servers) that need access to information on how endpoints are behaving while connected to the network; and access control enforcers that need access to endpoint state information and/or user identity to make connectivity decisions.

3.3 The TNC Access Control Capability

The Access Control capability decides which actions should be permitted, based on information consumed from the Compliance capability, the Orchestration capability, and/or locally configured access policies, and enforces these decisions. These decisions may be based on a variety of factors including endpoint identity, posture, behavior, and/or location; user identity and role; external assessments; and input from other security technologies. This enables the administrator to have a unified policy for access control, encompassing all security-relevant factors, across various services and network topologies (local and remote, wired and wireless, etc.).

When an endpoint requests an action, the Access Control capability consumes information about the endpoint and compares the results to security policy. Based on the policy, the Access Control capability decides whether the requested action should be permitted and enforces that decision. Enforcement may happen at the edge of the network (e.g. 802.1X-enabled switches and wireless access points, or VPN gateways), inline within the network (e.g. firewalls and proxies), or integrated into the application or service requested.

For example, when an endpoint attempts to connect to a network, the Access Control capability may gather information about the endpoint's identity and posture and decide whether to allow the endpoint to connect to the network, send it to a quarantine or remediation VLAN, or block the endpoint’s access all together. The Access Control capability then, via the enforcement point to which the endpoint is connected, provides appropriate access (or lack thereof) to the endpoint.

The Access Control capability can change an endpoint’s connectivity whenever network policy states that it is necessary. For example, an endpoint that was compliant to policy at the time it joined the network may fall out of compliance; the Access Control capability can then quarantine the endpoint.

3.4 Combining Capabilities

While each of the three above capabilities of the TNC Architecture are compelling in their own right, they can work together to provide a broad view of network health, and enable remediation of at-risk or infected endpoints. The strength of each capability is enhanced through integration, offering a real-time view of the dynamic security state of the enterprise. This bird's-eye view includes (but is not limited to) endpoint behavior, state, and location, and network activity and threats, enabling administrators and security devices to draw conclusions and send out or execute instructions for action in response to activity, policy changes, and emergent threats.

For example, if a network administrator chose to deploy tools implementing both the TNC Compliance capability and the TNC Orchestration capability, he would be able to compare what an endpoint reported to what an endpoint was observed doing on the network.

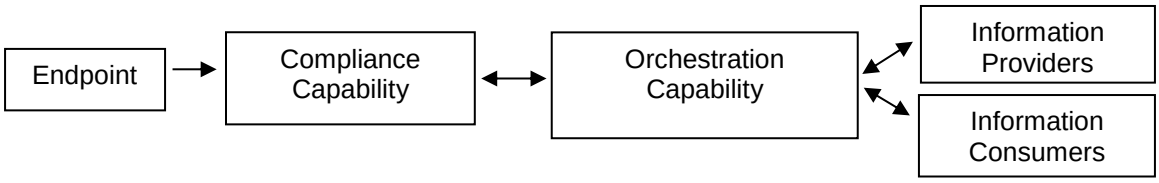

Figure 3: Combining the Compliance and Orchestration Capability

The Compliance capability can make compliance data available to the Orchestration capability.

The Orchestration capability can in turn share this data with the information consumers that have subscribed to the Orchestration capability for this type of data. This helps to enable several network operations, including (but not limited to):

- Sharing compliance data with consumers that can use it to make their own connectivity decisions
- Identify "lying endpoints" by comparing their behavior on the network to their compliance report
- Looking for similar configurations on endpoints that are demonstrating "bad behavior" on the network

Additionally, an Access Control capability in conjunction with either or both of the Compliance capability and an Orchestration capability can make much better network access control decisions then it can alone.

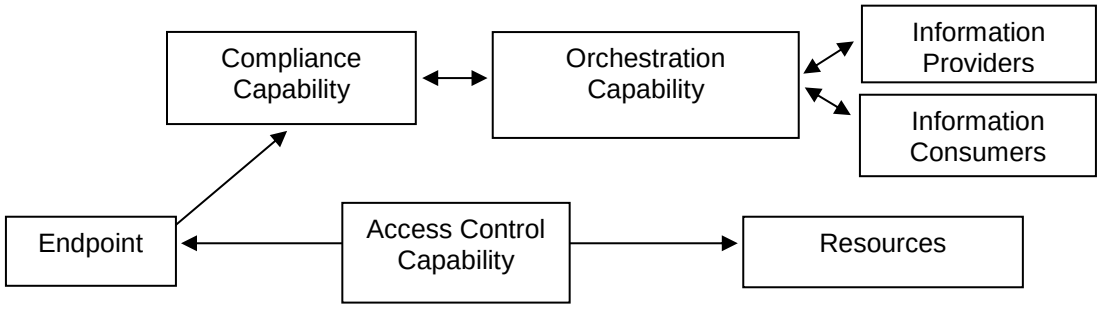

Figure 4: Combining the Compliance, Access Control, and Orchestration Capabilities

Without connections to a Compliance or Orchestration capability, an Access Control capability only has access to the information it can gather from an endpoint directly (for example, via 802.1X). However, when the Access Control capability is implemented with the Compliance or Orchestration capability, it can begin asking more intelligent questions about a connecting endpoint, such as:

- Is compliance data available for this endpoint? If so, was it compliant to network policy last time it was connected? Based on its last compliance report, is it compliant with the most recent network policy?
- An endpoint with a certain identity is attempting to connect to the network - do any of the network tools or access control enforcers have an opinion about this endpoint?

Answers to these and other questions allow the Access Control capability to make better decisions regarding an endpoint's request for access.

Once an endpoint has joined the network, the Compliance and Orchestration capabilities can make use of the Access Control capability's quarantine functionality. For example, if the Compliance capability decides that an endpoint's access must be restricted until it is updated, it can send a message to the Access Control capability, which can change the endpoint's access. Other infrastructure, including access control enforcers, can perform the same kind of quarantine based on coordination through the Orchestration capability.

These capabilities can be applied to a variety of usage scenarios as discussed in Section 4.

3.5 Composing the TNC Capabilities

Capabilities are enabled by the various components of the TNC Architecture playing their specific roles. These components, and the resulting capabilities, can be combined to deliver the use cases and usage scenarios outlined in this document, as well as additional use cases and usage scenarios that may be unique to a specific environment. The components work together to deliver the various capabilities, and the capabilities can be leveraged to deliver a variety of solutions.

The TNC roles described in section 2.1.3 map to the TNC capabilities described in this section. Each capability contains a minimum necessary set of roles; additional roles - as well as components not standardized by TNC - could also contribute to each capability.

The minimum roles required for the Compliance capability are:

- an endpoint (from which posture information is collected)
- a policy server (collecting the information, evaluating it, and storing it in a CMDB)
- a CMDB (storing posture information and providing it to authorized requestors)
- CMDB clients (consuming endpoint posture information and potentially acting upon it)

The minimum roles required for the Orchestration capability are:

- a MAP (storing and distributing shared information)
- MAP Clients (publishing, searching for, and subscribing to updates on information)

Other roles in the TNC Architecture may simultaneously act as MAP Clients (such as a policy server publishing information, a CMDB retrieving information, etc.).

The minimum roles required for the Access Control capability are:

- an endpoint (requesting access to a resource)
- an enforcement point (controlling access to the requested resource)
- a policy server (provisioning access control to the enforcement point)

As the capabilities are combined, multiple functions of a TNC Architecture entity may be leveraged in the service of those capabilities. For example, a policy server that evaluates the posture of an endpoint before allowing access to a resource is participating in both the Compliance capability (for posture evaluation) and the Access Control capability (for resource access control.)

4 Usage Scenarios

A usage scenario is a description of what a TNC user might want to do using Trusted Network Communications, and the context in which he or she wants to do it. TNC addresses a broad range of usage scenarios. The following sections describe only a few common scenarios; the TNC Architecture can be used in many scenarios beyond these examples.

4.1 Vulnerability Mitigation

A systems administrator received information about a newly discovered vulnerability in a popular piece of software. He needs to mitigate this vulnerability as quickly as possible.

To this end, the systems administrator determines which endpoints have this vulnerable software installed. He evaluates those endpoints for signs of attack, and removes endpoints that may have been attacked to a secure network enclave for investigation. He sends instructions to vulnerable endpoints to run their patch updater. He sets an alert to monitor those endpoints, in case they begin to demonstrate behavior that could indicate they were attacked. He creates a new policy that blocks network access from endpoints joining the network that are running the vulnerable software.

4.2 Boot-time Anomaly Detection

A systems administrator is concerned about the possibility of boot-time configuration and software changes, such as might be caused by a rootkit infestation, in her environment, and wants visibility into endpoints potentially compromised.

To this end, the systems administrator leverages a root of trust, such as the TPM, to collect cryptographically-proven boot-time measurements that can be compared against an expected baseline.

4.3 Dynamic Policy Application

A solution architect is updating the design of a network enclave. She has created a policy that will enable fine-grained access control decisions and compliance testing, and now needs to implement this policy.

To this end, the solution architect pushes the new policy out to the network, where it is applied to endpoints that join the network. She tests currently connected endpoints against the new policy, and sends update instructions to the endpoints that are found to be out of compliance with this new policy.

4.4 Continuous Monitoring and Mitigation

A systems administrator needs visibility into the compliance state of endpoints, and real-time notification when an endpoint deviates from acceptable configuration or behavior, without being overwhelmed by the amount and variety of reported information.

To this end, a solution implementer defines a set of policies for each group of endpoints commensurate with the endpoint's accepted activities and employs an automated system to monitor the endpoints and take action where appropriate.

Endpoints automatically deliver state information to a compliance service, which evaluates that information against appropriate policy, isolates and/or remediates uncompliant endpoints, and notifies administrators of potential issues. The service periodically reassesses endpoints to ensure that they are still conforming to policies; if an endpoint has fallen out of compliance, the service immediately takes appropriate steps.

The systems administrator only becomes involved if her attention is needed to resolve a problem, and non-problematic endpoints are allowed to continue operation without any need for human involvement.

4.5 Security Automation

A solution architect desires to minimize the need for human intervention in routine network operations, including basic data collection and consolidation, allowing personnel resources to focus on exceptions, threats, and incidents.

To this end, the solution architect integrates multiple disparate network and security technologies to enable rapid, dynamic, responsive adjustment of the network's security posture based on myriad factors such as endpoint identity, posture, behavior, type, and location; network activity and threats; and other input factors from a variety of sources.

5 Elements of the TNC Architecture

Previous sections describe the capabilities enabled by the architecture; this section talks about the roles and functions that comprise the architecture, and the layers at which the functions operate. As described in section 2.1.2, a function is an element performing a specific operation of the TNC Architecture, and a role is an actor with a specific purpose in the TNC Architecture, utilizing a particular set of functions.

The TNC Architecture is implemented across individual entities (physical or virtual systems). An entity may play one or more roles in the TNC Architecture, by implementing one or more functions of the TNC Architecture. These roles participate in one or more of the capabilities.

TNC defines a standard set of capabilities, and could also be used to deliver other capabilities. The TNC Architecture is a toolbox for delivering those capabilities. The TNC roles are different tools in that toolbox, and the TNC functions make up those tools (like the various blades on a pocketknife). You can use the tools described herein, with their standard combinations of functions, or build your own tools. Not all tools are useful for all scenarios, nor is it required to use all of the TNC tools to use TNC. The use of any combination of TNC components means you're using TNC.

5.1 TNC Architecture

The TNC Architecture, as standardized in the TNC specifications, is shown in Figure 2. The architecture is intentionally general due to the need for it to encompass a variety of network devices, topologies, and implementation configurations. The architecture incorporates several layers, roles, functions, and interfaces as illustrated by this figure.

The vertical groupings in this figure depict the eight *roles* in the TNC architecture: the Endpoint, (formerly called the Access Requestor), the Policy Enforcement Point (PEP), the Policy Decision Point (PDP), the Compliance Evaluation Point (CEP), the Configuration Management Database (CMDB), the CMDB Client (CMDBC), the Metadata Access Point (MAP), and the MAP Client (MAPC). Within each role, the boxes depict the *functions* within those roles. Three horizontal shaded *layers* are depicted grouping related functions, while the *interfaces* standardized by TNC are depicted by named lines. Further sections provide detailed descriptions of these layers (5.1.1), roles (5.2), functions (5.3), and interfaces (7.1).

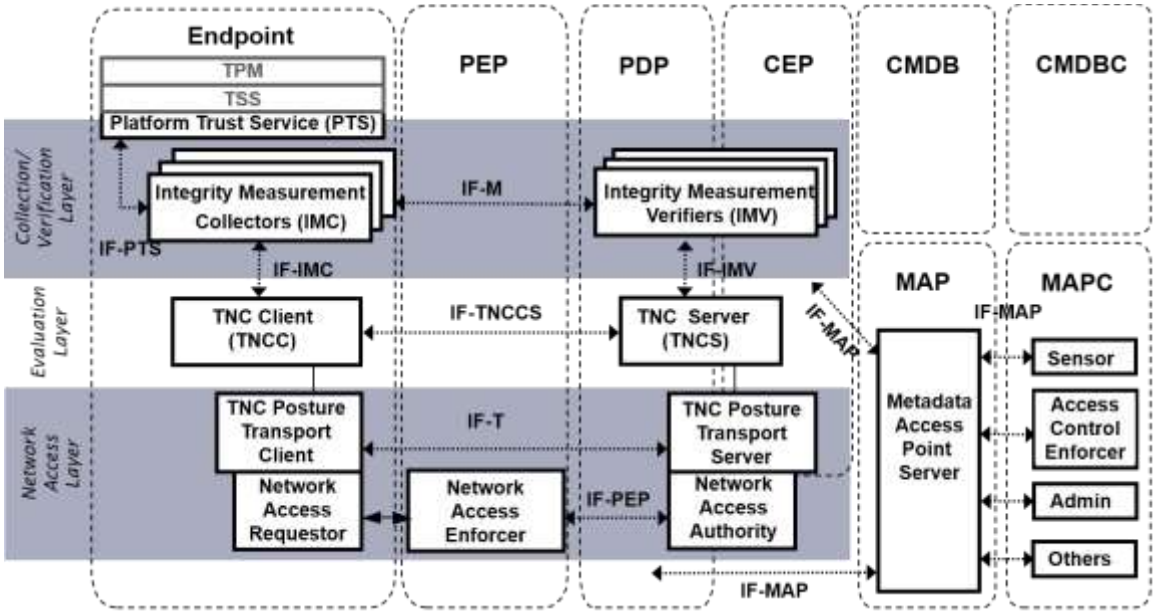

Figure 5: TNC Architecture Details

It is important to note that Figure 5 shows the functions of each role that pertain to the capabilities of TNC. The TNC Architecture does not preclude other components that implement other functions pertaining to compliance, access control, orchestration, and networking and security in general. For example, the Network Access Authority (NAA) could be implemented as just an additional component within a RADIUS Server within a given 802.1X usage, with the RADIUS Server also obtaining other policy-related information from other sources (e.g. other servers). As such, it is important for the reader to understand that the functions of each role in the TNC Architecture are not the only components implementing security, network connection management, and information exchange.

Additionally, a single entity in a network environment may play more than one role in the TNC Architecture. For example, a switch or wireless access point configured to authenticate endpoints with 802.1X supplicants via 802.1X, but assign a default access policy (e.g. guest VLAN) to non-supplicant endpoints, fulfills the role of both the PEP and PDP; such a network device is referred to as a combined PEP/PDP. Another example is a policy server that collects endpoint posture information and stores it in a CMDB, and subscribes to information from a MAP; that policy server is both a CEP and a MAPC.

All roles and functions in the architecture are logical ones, not physical ones. The entity performing a particular role or component providing a particular function may be a single software program, a hardware or virtual machine, or a redundant and replicated set of machines spread across a network, as appropriate for its function and for the deployment's needs. (The exception is a TPM, which is preferably a hardware component in order to guarantee robust TPM isolation. The TNC Architecture accommodates both endpoints with a TPM and those without; if a TPM is present on an endpoint, the TNC Architecture can leverage that TPM to increase trust in information collected from that endpoint.)

5.1.1 Layers

TNC architecture functions supporting endpoint compliance assessment (as defined in section 5.3) operate at three abstract layers of the architecture:

- *The collection/verification layer*: This layer contains plug-in components whose function is to collect and verify endpoint posture information. The functions found in this layer are the Integrity Measurement Collectors (IMCs) and Integrity Measurement Verifiers (IMVs).
- *The evaluation layer*: This layer contains components whose function is to evaluate the overall posture of the endpoint with respect to certain access and compliance policies, with input from the functions at the collection/verification layer. The functions found in this layer are the TNC Client (TNCC) and the TNC Server (TNCS).
- *The network access layer*: This layer contains components whose function pertains to traditional network connectivity and security. They may support a variety of networking technologies (e.g. VPN, 802.1X, TLS). The functions found in this layer are the TNC Posture Transport Client (TPTC), the TNC Posture Transport Server (TPTS), the Network Access Requestor (NAR), the Network Access Enforcer (NAE), and the Network Access Authority (NAA).

These layers are useful for illustrating the pairings of functions used in endpoint compliance assessment - for example, IMCs with IMVs, TNC Clients with TNC Servers, etc.

5.2 Roles

TNC currently defines the following set of roles in the TNC architecture: the Endpoint (formerly called the Access Requestor), the Compliance Evaluation Point (CEP), the Configuration Management Database (CMDB), the CMDB Client (CMDBC), the Policy Decision Point (PDP), the Policy Enforcement Point (PEP), the Metadata Access Point (MAP), and the MAP Client (MAPC).

These roles are standardized by TNC at this time; however, other roles may be constructed using the functions defined by TNC (see section 5.3).

5.2.1 Endpoint

The role of the Endpoint in the TNC Architecture is to provide posture information to a Policy Server, and to access a protected resource. The Endpoint may be in the process of connecting to the network, or may already be connected to the network.

An Endpoint is the subject of the various capabilities.

5.2.2 Policy Enforcement Point (PEP)

The role of the PEP in the TNC Architecture is to enforce the decisions of the PDP regarding resource access. (Note that other components of a TNC-enabled environment, such as an IMC, may enforce other policy decisions, such as collection policy, using other mechanisms.)

A PEP is primarily associated with the Access Control capability because it consumes policy decisions from the PDP and enforces access control policy.

5.2.3 Policy Server

Two roles in the TNC Architecture are related variants of the same concept - a server that applies policy to some set of actions, or Policy Server. Those two roles are the Policy Decision Point (PDP) and the Compliance Evaluation Point (CEP).

5.2.3.1 Policy Decision Point (PDP)

The role of the PDP in the TNC Architecture is to perform the decision-making regarding the Endpoint's resource access request, in light of the access policies. The PDP compares the Endpoint's credentials (e.g. user certificates, password, etc.) and information about its posture against configured network access policies and then decides whether network access should be granted to the Endpoint. If a PEP is present, the PDP then communicates its decision to the PEP.

A PDP is primarily associated with the Access Control capability, because it makes access control decisions and provisions access control policy to the PEP, and the Compliance capability, because it gathers compliance information which it uses in its decision-making process. It can communicate with the Orchestration capability to share endpoint access information and consume information from other sources as additional inputs into its decision-making process.

5.2.3.2 Compliance Evaluation Point (CEP)

The role of the CEP in the TNC Architecture is to collect and evaluate endpoint posture information, and to send the information to a CMDB for storage. This information can then be made available to other CMDB Clients.

A CEP is primarily associated with the Compliance capability because it gathers compliance information, and it can communicate with the Orchestration capability to share that information and consume compliance information from other sources.

5.2.4 Configuration Management Database (CMDB)

The role of the CMDB in the TNC Architecture is to store endpoint posture information and make it available to consumers of such information.

A CMDB is primarily associated with the Compliance capability because it stores compliance information, and it can communicate with the Orchestration capability to share that information and consume compliance information from other sources.

5.2.5 CMDB Client (CMDBC)

The role of the CMDB Client in the TNC Architecture is to share and consume endpoint posture information.

A CMDBC is primarily associated with the Compliance capability because it shares and consumes compliance information.

5.2.6 Metadata Access Point (MAP)

The role of the MAP in the TNC Architecture is to store and provide state information about endpoints, resources, and the environment. This information may include, but is not limited to, device bindings, user bindings, registered address bindings, authentication status, endpoint compliance status, endpoint behavior, and authorization status.

A MAP is primarily associated with the Orchestration capability because it coordinates sharing of information among various components.

5.2.7 MAP Client (MAPC)

The role of the MAP Client in the TNC Architecture is to publish to, or consume from, the state information in the MAP. A MAP Client may both publish and consume state information.

A MAP Client is primarily associated with the Orchestration capability because it publishes and consumes information shared with other components.

5.3 Functions

Referring to Figure 5, the functions making up the roles are as follows.

5.3.1 Endpoint Functions

The Endpoint has the following TNC-specific functions, which are generally implemented as software components running on the Endpoint:

- *Integrity Measurement Collectors (IMCs)*: The IMC function measures specific aspects of the Endpoint's posture.[2] Example measurements could include the anti-virus parameters on the Endpoint, personal firewall status, software and operating system versions, patch levels, configuration, and other provisioning and security aspects of the Endpoint. Endpoint posture, the overall state of the Endpoint, is the aggregate of all of the individual posture measurements. Note that the TNC Architecture is designed to enable multiple IMCs to interact with a single, or multiple, TNCC(s) and TNCS(s), thereby allowing customers to deploy sophisticated compliance policies involving a range of vendors' products. An IMC is interoperable with a NEA Posture Collector [1]; IMCs may also have remediation functionality.
- *TNC Client (TNCC)*: The TNCC function receives requests and instructions from a TNC Server (see 5.3.2), sends them to the appropriate IMCs, combines posture measurements from IMCs into batches, and communicates them to a TNCS. An instance of this transfer is referred to as an Integrity Check Handshake. A TNCC is interoperable with a NEA Posture Broker Client [1].
- *TNC Posture Transport Client (TPTC)*: The TPTC function facilitates network communication to a TNC Posture Transport Server (see 5.3.2), over which the TNCC

[2] IMCs originally were focused on integrity measurements – information that would indicate whether a device has been compromised. Now IMCs can collect endpoint posture information for a number of other use cases.

communicates posture information. [3] A TPTC is interoperable with a NEA Posture Transport Client [1].

- *Network Access Requestor (NAR)*: The NAR is the function responsible for establishing network access by negotiating an Endpoint's connection to a network. There may be several NARs on a single Endpoint to handle connections to different networks. The supplicant in 802.1X or the VPN client in IPsec are examples of NARs. This function is not used when communicating with a Compliance Evaluation Point (see 5.3.2), but is used when communicating with a Policy Decision Point (see 5.3.5).

5.3.2 Compliance Evaluation Point Functions

The Compliance Evaluation Point has the following TNC-specific functions:

- *Integrity Measurement Verifiers (IMVs)*: The IMV function compares a particular aspect of the Endpoint's posture against policy, based on measurements received from IMCs or other data, and returns an IMV Action Recommendation. The IMV function on a CEP may also communicate between the CEP and the CMDB. An IMV is interoperable with a NEA Posture Validator [1]; IMVs may also send remediation instructions.
- *TNC Server (TNCS)*: The TNCS function collects requests and instructions from IMVs, communicates them to a TNCC, receives batches of measurements from the TNCC, distributes the measurements to the appropriate IMVs, and gathers IMV Action Recommendations from IMVs. The TNC Server is interoperable with a NEA Posture Broker Server [1].
- *TNC Posture Transport Server (TPTS)*: The TPTS function facilitates network communication to a TPTC. Information used by an Endpoint to authenticate to the TPTS may be used to uniquely identify the Endpoint to the rest of the CEP functions.[4] A TPTS is interoperable with a NEA Posture Transport Server [1].

In most cases, the CEP will also include a AAA server for endpoint authentication. One primary distinction between a CEP and a PDP (see 5.3.5) is that the PDP has a Network Access Authority and performs enforcement functions, whereas a CEP performs authentication but not enforcement.

5.3.3 CMDB and CMDB Client Functions

The CMDB stores endpoint information, including identity and posture, collected by the CEP and by other CMDB Clients from endpoints on the network. The CMDB may be local to the CEP or remote.

Examples of CMDB Clients include asset management, vulnerability analysis, and threat detection systems, and other tools that collect information and store it in the CMDB, or consume information from the CMDB in order to perform their intended function.

No TNC-specific functions are currently defined for these roles. They are included in the TNC Architecture to represent generation and storage/retention of endpoint information.

[3] In previous versions of the TNC Architecture, the TPTC was not a separate component – its roles were covered by the NAR. However, separating the two roles allows posture transport to take place independent of network access requests, which supports several TNC use cases.

[4] In previous versions of the TNC Architecture, the TPTS was not a separate component – its roles were covered by the NAA. However, separating the two roles allows posture transport to take place independent of network access requests, which supports several TNC use cases.

5.3.4 Policy Enforcement Point Functions

The PEP has the following TNC-specific function:

- *Network Access Enforcer (NAE)*: The NAE function controls access to a protected network. The NAE consults an NAA to determine whether this access should be granted. One example of the NAE is the Authenticator in 802.1X, which is often implemented within an 802.11 switch or access point.

5.3.5 Policy Decision Point Functions

The PDP has the following TNC-specific functions:

- *Integrity Measurement Verifiers (IMVs)*: See section 5.3.2.
- *TNC Server (TNCS)*: See section 5.3.2.
- *TNC Posture Transport Server (TPTS):* See section 5.3.2.
- *Network Access Authority (NAA)*: The NAA function decides whether an Endpoint should be granted access. The NAA consults a TNC Server to determine whether the Endpoint's posture complies with the PDP's security policy. In many cases, an NAA will be implemented within a AAA Server running on the PDP, but this is not required.

5.3.6 Metadata Access Point Functions

The MAP has the following TNC-specific function:

- *Metadata Access Point Server (MAPS)*: The MAPS function is a component to which other TNC components may publish, subscribe, and search data which reflects the state of TNC elements and aids in decision making and policy enforcement. The MAPS allows components in addition to PEPs, such as Access Control Enforcers, to enforce policies based on relationships to endpoints, users, capabilities, roles, device activities and postures as well as other run time data. The MAPS allows elements which do not have a direct relationship with an Endpoint, such as Sensors, to publish observed or collected information about Endpoint activities which may be of interest to PEPs, PDPs/CEPs, and other MAP Clients.

A MAP Service may be comprised of a single MAP or may be distributed across multiple MAPs.

5.3.7 MAP Client Functions

Examples of MAP Client functions include:

- *Administrative Client*: The Administrative Client function enables administrative operations such as monitoring, investigation, and provisioning by utilizing information from the MAP and publishing information to the MAP via IF-MAP. Examples of Administrative Clients include data visualizers, configuration management databases (CMDBs), Policy Information Points (PIPs), and provisioning servers. Examples of operational activities enabled include data exploration, asset management, and certificate lifecycle management.
- *Access Control Enforcer (ACE)*: The Access Control Enforcer (formerly Flow Controller) function makes and enforces decisions about network access utilizing information from the MAP. ACEs take action (e.g. block) on network activities (i.e. network traffic associated with a particular endpoint, device, user, etc.) based on data obtained via IF-MAP. Examples of ACEs include VPN gateways, internal firewalls, inline intrusion prevention systems (IPSs), rate limiters, and proxies. Examples of network activities being controlled include accessing particular services in a network, accessing particular geographies in a network, and restricting the amount of bandwidth allowed.

An Access Control Enforcer differs from a PEP in that a PEP enforces policy decisions provisioned from PDP, whereas an Access Control Enforcer makes and enforces policy decisions based on information consumed via IF-MAP.

- *Back-Haul Interface* (BHI): "Backhaul Interface" (BHI) MAP Clients segregate protected ICS network devices communicating across a shared, untrusted commodity IP infrastructure. Virtual overlay networks can be used to protect communications across so-called "backhaul networks", providing connectivity between ICS components over some distance. The BHI implements necessary authentication, encryption, translation, and authorization policy enforcement capabilities to create the overlay network, using IF-MAP for coordination, provisioning, and management.
- *Sensor*: The Sensor function monitors the environment; gathers information such as network activities, location, and other observations about Endpoints; and publishes information to the MAP via IF-MAP. Examples of Sensors include intrusion detection devices, network virus detection devices, layer 3 traffic monitors, application traffic scanners, location awareness systems, vulnerability assessment. Examples of network activities being monitored include accessing particular services in a network, authentication activity, broadcast requests for various services (e.g. DHCP), and advertising of services. Additional information such as location, physical characteristics, posture, etc. may also be observed by Sensors.

  In addition to publishing information, a Sensor may also consume information, such as a request for information from another MAP Client, or information that the Sensor uses to determine which policies to apply to its information collection activities.

Many other functions of MAP Clients may be imagined and/or implemented; the previous set represents only those playing a specific role in current TNC specifications.

5.4 Flexibility of the TNC Architecture

The TNC Architecture is implemented across individual entities (physical or virtual systems). An entity may play one or more roles in the TNC Architecture. These roles participate in one or more of the capabilities. TNC has defined a standard set of roles to support these capabilities; these roles can be combined in various ways to support the different capabilities of the TNC architecture. The capability is a benefit delivered by a set of roles, but those roles are not fixed. For example, a PDP role, a CEP role, and a CMDB role can all contribute to the Compliance capability; however, the Compliance capability does not require all of those roles to be implemented. An implementation incorporating a CMDB with a CEP - but not a PDP - can still leverage benefits of the Compliance capability.

Furthermore, an implementer or product developer may combine the functions of the TNC architecture differently to create entirely new roles. For example, an implementer may create an entity which only implements an IMV, without any other functions - that entity could play a role of a remote IMV server. That role is not a standard role within the TNC Architecture, but could serve a useful purpose in a TNC ecosystem.

6 Applying the TNC Architecture

Working in combination, the elements of the TNC Architecture enable a variety of use cases. These use cases are functional building blocks that can be combined to deliver solutions in the usage scenarios described in section 4.

6.1 Endpoint Identification Verification

Description:	An Endpoint reports its identity to a Policy Server, which validates the identity.
Capability:	Compliance and Access Control
Actors:	Endpoint (5.3.1) and Policy Server (5.2.3)
Primary Success Path:	1. An Endpoint connects to a Policy Server and sends identity credentials, or identity credentials are sent to the Policy Server on behalf of the Endpoint (as in MAC-based RADIUS authorization). 2. The Policy Server checks the identity against cryptographic validator or back-end authentication data store. 3. The Policy Server decides whether to accept the identity presented by the Endpoint. 4. The Policy Server optionally re-validates the identity at predefined points in the future.

6.2 Endpoint Posture Collection - Server Initiated

Description:	A Policy Server collects up-to-date posture information from an Endpoint.
Capability:	Compliance
Actors:	Endpoint and Policy Server
Primary Success Path:	1. A posture check is initiated by a Policy Server. Communication is established either over a new or existing connection. 2. The Policy Server sends a request for information to the Endpoint via the TNC Server (5.3.2) on the Policy Server and the TNC Client (5.3.1) on the Endpoint. 3. One or more IMCs (5.3.1) on the Endpoint receives and process the request. 4. The requested information is communicated back to the corresponding IMV(s) (5.3.2) on the Policy Server. The Policy Server consumes the information.

6.3 Endpoint Posture Collection - Endpoint Initiated

Description:	An Endpoint sends up-to-date posture information to a Policy Server.
Capability:	Compliance
Actors:	Endpoint and Policy Server
Primary Success Path:	1. A posture check is initiated by an Endpoint. Communication is established either over a new or existing connection.

	2. One or more IMCs on the Endpoint gathers the information and sends it via the Endpoint's TNC Client to a Policy Server's TNC Server. 3. The requested information is received by the corresponding IMV(s) on the Policy Server. The Policy Server consumes the information.

6.4 Endpoint Posture Evaluation

Description:	A Policy Server evaluates collected posture information.
Capability:	Compliance
Actors:	Policy Server
Primary Success Path:	1. One or more IMVs compares collected posture information to applicable policy. 2. Each IMV makes a recommendation to a TNC Server. 3. The TNC Server aggregates the IMVs' recommendations to produce an evaluation result.

6.5 Endpoint Posture Information Storage

Description:	A Policy Server sends collected posture information to a CMDB for storage.
Capability:	Compliance
Actors:	Policy Server and CMDB (5.2.4)
Primary Success Path:	1. An IMV on a Policy Server sends collected posture information to a CMDB, which is either co-located with the Policy Server or is hosted separately. 2. The CMDB consumes and stores the posture information.

6.6 Endpoint Remediation

Description:	An Endpoint acts upon a remediation request sent from a Policy Server.
Capability:	Compliance and Access Control
Actors:	Policy Server and Endpoint
Primary Success Path:	1. A Policy Server sends a remediation instruction to an IMC for an action. 2. The IMC performs the action and reports back on its success to the Policy Server.

6.7 Access Control Decision

Description:	A Policy Server compares Endpoint information against configured policy and makes an access control decision.
Capability:	Access Control

Actors:	Policy Server
Primary Success Path:	1. A Policy Server evaluates Endpoint identity and collected posture information and compares it to the policy configured on the Policy Server. 2. The Policy Server makes an access control decision based on whether the Endpoint is compliant with the appropriate policy.

6.8 Access Control Enforcement

Description:	An Endpoint requests access to a resource, and an Enforcement Point consults a Policy Server to determine whether to grant that access.
Capability:	Access Control
Actors:	Endpoint, Enforcement Point (5.2.2), and Policy Server
Primary Success Path:	1. An Endpoint requests access to a resource protected by an enforcement point. 2. The Enforcement Point consults a Policy Server to determine whether to grant access. 3. The Enforcement Point grants or denies access based on response from the Policy Server. 4. The Enforcement Point optionally re-evaluates access granted based on future input.

6.9 Information Publication

Description:	A MAP Client provides information to help build a repository of information.
Capability:	Orchestration
Actors:	MAP Client (5.2.7) and MAP Service (5.2.6)
Primary Success Path:	1. A MAP Client connects to a MAP Service. 2. The MAP Client publishes information. 3. (Optional) The MAP Service validates the information. 4. The MAP Service adds the information to its repository.

6.10 Information Search and Consumption

Description:	A MAP Client searches for information in a repository and consumes that information.
Capability:	Orchestration
Actors:	MAP Client and MAP Service
Primary Success Path:	1. A MAP Client connects to a MAP Service. 2. The MAP Client searches for desired information. 3. The MAP Service sends back information matching the search. 4. The MAP Client consumes the returned information.

6.11 Information Alerting

Description:	A MAP Client subscribes to receive updates to information and consumes updated information.
Capability:	Orchestration
Actors:	MAP Client and MAP Service
Primary Success Path:	1. MAP Client A connects to a MAP Service. 2. MAP Client A subscribes to receive updates on desired information. 3. The MAP Service sends back existing information matching the subscription. 4. MAP Client B connects to the MAP Service. 5. MAP Client B publishes new information matching MAP Client A's subscription. 6. The MAP Service notifies MAP Client A of new information from MAP Client B matching the existing subscription. 7. MAP Client A consumes the returned information.

6.12 Request for Collection of Information

Description:	A MAP Client requests information; the MAP Service brokers collection of that information and returns it to the MAP Client, which consumes the information.
Capability:	Orchestration
Actors:	MAP Client and MAP Service
Primary Success Path:	1. MAP Client A connects to a MAP Service. 2. MAP Client A subscribes to be notified when certain information is requested. 3. MAP Client B connects to the MAP Service. 4. MAP Client B publishes a request for information and subscribes to receive updates matching the requested information. 5. The MAP Service notifies MAP Client A of the request for information. 6. MAP Client A acts on the request for information and publishes resulting information. 7. The MAP Service notifies MAP Client B of new information from MAP Client A matching the existing subscription. 8. MAP Client B consumes the returned information.

7 TNC Specifications

The following TNC specifications define the interfaces, profiles, and schema used in Trusted Network Communications:

- TNC Architecture for Interoperability
- IF-IMC - Integrity Measurement Collector Interface
- IF-IMV - Integrity Measurement Verifier Interface
- IF-TNCCS - Trusted Network Connect Client-Server Interface
  - IF-TNCCS: TLV Binding
  - IF-TNCCS: Protocol Bindings for SoH
- IF-M - Vendor-Specific IMC/IMV Messages Interface
  - IF-M: TLV Binding
  - SWID Message and Attributes for IF-M
  - Attestation PTS Protocol: Binding to IF-M
  - IF-M Segmentation
- IF-T - Network Authorization Transport Interface
  - IF-T: Protocol Bindings for Tunneled EAP Methods
  - IF-T: Binding to TLS
- IF-PEP - Policy Enforcement Point Interface
  - IF-PEP: Protocol Bindings for RADIUS
- IF-MAP - Metadata Access Point Interface
  - IF-MAP Binding for SOAP
  - IF-MAP Metadata for Network Security
  - IF-MAP Metadata for ICS Security
  - MAP Content Authorization
- ECP - Endpoint Compliance Profile
- CESP - Clientless Endpoint Support Profile
- Server Discovery and Validation
- Federated TNC
- IF-PTS - Platform Trust Services Interface
  - Simple Object Schema
  - Core Integrity Schema
  - Integrity Report Schema
  - Reference Manifest (RM) Schema
  - Security Qualities Schema
  - Verification Result Schema

These are the current specifications as of the publication of this document; for the most up-to-date set of specifications, consult https://www.trustedcomputinggroup.org/work-groups/trusted-network-communications/open-standards-tnc/.

7.1 TNC Interfaces

A number of interfaces are shown in Figure 5, defining relationships between functions and the protocols and messages exchanged between functions. These interfaces are briefly discussed here.

7.1.1 Relationship with IETF NEA

A few years after the TNC workgroup started work on endpoint compliance standards, the IETF also began work in this space, through their Network Endpoint Assessment (NEA) workgroup. In order to avoid divergent standards, the TNC WG submitted a subset of the TNC endpoint compliance standards as candidate protocols to IETF NEA. These TNC standards were

accepted as a starting point for IETF protocol work, and a productive development process in NEA resulted in improvements to the standards, which the TNC WG then reflected back in updated TNC specifications. For more information on IETF NEA, see RFC 5209 [1] which provides an overview of NEA (similar to this document for the TNC ecosystem).

IETF NEA addresses only the Compliance capability aspect of TNC; the TNC Access Control and Orchestration capabilities are out of scope for NEA. As a result, not every TNC role or component has a NEA equivalent, and not every TNC standard has a NEA counterpart.

Today, four of the TNC endpoint compliance standards (IF-M, IF-TNCCS, IF-T/EAP, and IF-T/TLS) have equivalent, interoperable NEA RFC counterparts, as follows:

TCG TNC	IETF NEA
IF-M: TLV Binding 1.0 [10]	RFC 5792 - PA-TNC [5]
IF-TNCCS: TLV Binding 2.0 [11]	RFC 5793 - PB-TNC [6]
IF-T Binding to TLS 2.0 [12]	RFC 6876 - PT-TLS [7]
IF-T: Protocol Bindings for Tunneled EAP Methods 2.0 [13]	RFC 7171 - PT-EAP [8]

NEA uses slightly different terminology to refer to several of the concepts addressed by TNC. The following table illustrates the relationship between TNC and NEA terms:

TNC Term	NEA Term
Integrity measurement	Posture Attribute
Endpoint	NEA Client
Integrity Measurement Collector (IMC)	Posture Collector
TNC Client (TNCC)	Posture Broker Client
TNC Posture Transport Client (TPTC)	Posture Transport Client
Network Access Requestor (NAR)	[No equivalent]
Compliance Evaluation Point (CEP)	NEA Server
Integrity Measurement Verifier (IMV)	Posture Validator
TNC Server (TNCS)	Posture Broker Server
TNC Posture Transport Server (TPTS)	Posture Transport Server
Policy Decision Point (PDP)	NEA Server + policy enforcement
Network Access Authority (NAA)	[No equivalent]

The benefits of collaboration between TCG and other standards groups are significant: broader visibility, improved standards, and greater uniformity in the industry-wide standards ecosystem.

7.1.2 Integrity Measurement Collector Interface (IF-IMC)

IF-IMC is the interface between Integrity Measurement Collectors (IMCs) and a TNC Client (TNCC). IF-IMC enables message exchanges between the IMCs and the IMVs, primarily used to gather information from IMCs so it can be communicated to IMVs. It also allows IMCs to

coordinate with the TNC Client as needed. For more details about IF-IMC, refer to the IF-IMC Specification [14].

Software, firmware, and hardware components are expected to report status information to the TNC Client on the Endpoint. The TNC Client supports an API to allow these components to communicate with it locally to report component-specific status information. The TNC Client acts as a conduit for the IMC that collects information from possibly multiple software, firmware, and hardware components, and delivers the information to the peer IMV through the TNC Server.

7.1.3 Integrity Measurement Verifier Interface (IF-IMV)

IF-IMV is the interface between IMVs and a TNC Server (TNCS). IF-IMV enables message exchanges between the IMCs and the IMVs, primarily used to deliver information sent from client-side IMCs to corresponding IMVs, and allows IMVs to supply their recommendations to the TNCS. For more details about IF-IMV, refer to the IF-IMV Specification [15].

7.1.4 IMC-IMV Messaging Interface (IF-M)

IF-M pertains to a message exchange that may occur between IMCs and IMVs; the messages may be either standard or vendor-specific. These messages are identified by a message type with an allocation system designed to avoid accidental reuse of types. These messages are carried over the IF-TNCCS interface. The TNC has standardized certain widely useful IF-M messages, such as IF-M protocol bindings for TLV [10] (which can express basic endpoint information), PTS binding for IF-M [16], and SWID Message and Attributes for IF-M [17], and may standardize additional messages.

Note that both IF-TNCCS and IF-M are relevant not only to trusted network communications, but to the larger TCG requirements around platform management.

The IF-M Segmentation specification [18] provides a standard means to manage the size of IF-M messages between TNC clients and servers. It also provides a mechanism by which large messages can be delivered in segments, to avoid overwhelming the network connection and/or the memory capacity of either the client or the server.

7.1.5 TNC Client-Server Interface (IF-TNCCS)

IF-TNCCS relates to interaction between the TNCC and the TNCS as it pertains to the exchange of endpoint information. More specifically, this interface defines a protocol that conveys:

- Messages from IMCs to IMVs (such as batches of posture information)
- Messages from IMVs to IMCs (such as requests for additional posture information, or remediation instructions)
- Session management messages, as they pertain to (a) and (b) above, and other session synchronization information between the TNCC and TNCS.

Note that the contents of the messages being passed between the IMCs and IMVs are opaque to the IF-TNCCS layer. IF-TNCCS relies on the underlying network authorization transport protocol (IF-T) to provide a secure authenticated channel to protect the messages in transit between the TNCC and the TNCS, and ensure they are delivered to the correct TNCC or TNCS.

Several protocol bindings for IF-TNCCS have been released: the original XML version of IF-TNCCS [19], IF-TNCCS-SOH [20], and TLV Binding for IF-TNCCS 2.0 [11]. The different feature sets of these protocols are the reason to have these alternate protocols. The TLV Binding for IF-TNCCS 2.0 is interoperable with NEA PB-TNC [6].

7.1.6 TNC Network Transport Interface (IF-T)

IF-T pertains to the transportation of messages between the TNC Posture Transport Server (TPTS) and the TNC Posture Transport Client (TPTC), handling the network-level communications between the Endpoint and the CEP or PDP including authentication, integrity,

and confidentiality for message transmission. TNC provides IF-T protocol bindings for Tunneled EAP Methods [13], which is interoperable with NEA PT-EAP [8], and TLS [12], which is interoperable with NEA PT-TLS [7].

7.1.7 Platform Trust Services Interface (IF-PTS)

IF-PTS provides an interface to hardware-based integrity measurements, also known as platform trust services, to assess whether TNC components are trustworthy. See section 8.5 and refer to the IF-PTS Specification [21] and PTS protocol bindings for IF-M [16] for additional details. IF-PTS has several supporting schema: Simple Object Schema [22], Core Integrity Schema [23], Integrity Report Schema [24], Reference Manifest (RM) Schema [25], Security Qualities Schema [26], and Verification Result Schema [27].

7.1.8 Policy Enforcement Point Interface (IF-PEP)

IF-PEP allows the PDP to communicate with the PEP, especially allowing the PDP to instruct the PEP to isolate the Endpoint during remediation and later grant it full network access once remediation is complete. For more details about IF-PEP, refer to the IF-PEP Specification [28].

7.1.9 Metadata Access Point Interface (IF-MAP)

IF-MAP allows elements in the TNC architecture to share and correlate stateful runtime metadata such as relationships of TNC components to endpoints, users, capabilities, roles, and attributes. IF-MAP provides publish, subscribe, and search interfaces between MAP Clients and the MAP. The data published and available via IF-MAP augments other sources of data for security-related decision making. Searches and subscriptions using IF-MAP return data which reflects recent metadata values and relationships reported by MAP Clients. For more details about IF-MAP, refer to the IF-MAP Binding for SOAP Specification [29], IF-MAP Metadata for Network Security Specification [30], IF-MAP Metadata for ICS Security Specification [31], and MAP Content Authorization Specification [32].

7.2 TNC Support Profiles

The TNC family of specifications includes profiles for aspects of trusted network communications that use existing interfaces in the TNC Architecture to enable specific tasks. These profiles do not define new interfaces; rather they describe solutions to real world problems faced by TNC Architecture customers.

7.2.1 Endpoint Compliance Profile

The Endpoint Compliance Profile describes a profile of TNC standards and capabilities that is optimized to collect Endpoint identity and posture attributes, and store this information in a searchable repository. This enables better awareness of the health of the entire enterprise by making it easier to perform analysis and investigation of the state of each Endpoint. The ECP makes it possible to share collected data with authorized applications and users, enabling analysis and correlation of enterprise state both past and present. This allows the data to be more easily used for enterprise-wide asset management, threat defense, and security management. For more details, refer to the ECP Specification [33].

7.2.2 Clientless Endpoint Support Profile

In today's environments, many endpoints exist that do not - or cannot - run a TNC Client, and therefore cannot provide posture information, yet still require access to a protected network. In the TNC approach, an endpoint without a TNC Client is defined as a Clientless Endpoint (CE).

Clientless Endpoints are addressed by the Clientless Endpoint Support Profile (CESP), which outlines an approach and enforcement mechanisms to ensure interoperability and enforce compliance in environments where some endpoints lack a TNC Client. There should be no expectation that the CESP will provide the same level of security provided for endpoints with

clients; the goal is to increase the ability of network operators to provide security for environments that contain Clientless Endpoints. For more details, refer to the CESP Specification [34].

7.3 Federated TNC

TNC standards specify how to assess the security posture of an Endpoint as it connects to the network. This assessment is performed by a TNCS belonging to the same security domain as the Endpoint; there exists a direct trust relationship between the Endpoint and the TNCS.

This trust relationship is sufficient provided that the Endpoint only accesses services within its own security domain. Federated TNC [35] addresses how the Endpoint's posture should be assessed by a service within other security domains. This specification defines how an Endpoint's posture can be queried and supplied such that a security domain, other than the Endpoint's own, can make authorization decisions controlling that Endpoint's access to its networks and applications.

7.4 Server Discovery and Validation

This specification provides a standard means by which an Endpoint may discover the presence of various types of TNC and related servers and determine whether those servers are recognized and thus suitable for interaction. A PDP is one example of such a server, but the specification supports many server types, including servers associated with the IETF NEA standards, as well as vendor-proprietary server types. As such, TNC Server Discovery and Validation [36] allows dynamic creation of relationships between an Endpoint and the servers with which it must interact.

8 TNC Architecture with the Trusted Platform Module

A TPM (Trusted Platform Module) is a system component that has state that is separate from the system on which it reports (the host system) [37]. The TPM can securely store artifacts used to authenticate a platform, such as a PC, laptop, or mobile phone. These artifacts can include passwords, certificates, or encryption keys. A TPM can also be used to store platform measurements that help ensure that the platform remains trustworthy. Authentication (ensuring that the platform can prove that it is what it claims to be) and attestation (a process helping to prove that a platform is trustworthy and has not been breached) are necessary steps to ensure safer computing in all environments.

The TNC Architecture accommodates platforms that have a TPM as well as those that do not. This section further delves into the details of the TNC Architecture for platforms that possess a TPM.

TNC in combination with TPM unites Platform Credential Authentication (using the TPM-related certificates) and Integrity Check Handshake (using the Platform Configuration Registers (PCRs) within the TPM) and makes them accessible to a Policy Server for use in endpoint evaluation, network access decisions, etc. Together, these two aspects comprise Platform-Authentication.

8.1 Benefits of TNC with TPM

Software is inherently untrustworthy; it can be compromised by malicious actors such that it no longer accurately reports endpoint integrity and/or posture. To generate trust in an Endpoint and the software running on it, the user needs a way to root that trust in something more reliable than software.

TNC provides a connection between measurements stored in the TPM and policy servers seeking to make decisions based on endpoint trustworthiness. Adding a TPM to the TNC architecture enhances trust in the identity and measurements that TNC collects from an Endpoint.

Policy servers have the ability to decide when it is safe to extend the enterprise boundary to a connecting platform based on the *integrity information* reported by the platform and by the *proof-of-identity* supplied by the platform. TPM increases the trustworthiness of both identity and integrity information.

TNC originally focused on *integrity measurements,* primarily around integrity of the BIOS / operating system and security controls (such as current anti-virus signature file or running endpoint firewall); TNC measurements have since expanded to include broader *posture measurements*, such as endpoint provisioning, installed software, and/or configuration. For the purposes of this section, "integrity" may refer to either or both type of measurement.

In the context of endpoint authentication and authorization, the aim is to ascertain the security state of a given platform or device. A strong hardware-protected root-of-trust is needed to ensure malware and improperly configured software can be detected if they report erroneous status (i.e. the "lying endpoint problem"). Use of a TPM can help ensure the trustworthiness of the data being fed to the TNC IMCs, by attesting to the trustworthiness of the platform providing the data.

8.2 Features of a Platform with a TPM

One of the core value propositions of the TNC approach is that a hardware protected root-of-trust within devices or platforms can help establish remote attestation of the integrity of the platform, and to communicate *platform proof-of-identity* (Platform Credential Authentication) and *platform integrity information* (Integrity Check Handshake) as part of an authentication event to an authentication server (PDP or IMV). Trust in a platform is built bottom-up, starting at the base with Trusted Platform Module (TPM) hardware bound to the platform’s motherboard.

An important concept that distinguishes the TCG approach to Platform-Authentication is the notion of a trusted platform containing a TPM that features *protected capabilities, attestations,*

integrity measurement and storage, and *integrity reporting*. All four properties or functions are core to trusted computing. These features are as follows:

1. *Protected Capabilities*: Protected capabilities are a set of commands with exclusive permission to access *Shielded Locations*. Shielded locations are places (memory, register, etc.) where it is safe (e.g. unavailable to malware code running on the CPU) to operate on sensitive data. The TPM implements protected capabilities and shielded locations. Among others, it is used to protect and report aggregations of integrity measurements that are stored inside the TPM's *Platform Configuration Registers* (PCRs). The TPM also stores cryptographic keys used to authenticate reported measurements. Depending on the platform and its implementation, TPM protected capabilities can include additional security functionality such as cryptographic key management, random number generation, sealing data to system state, and monotonic counters.
2. *Attestations*: Attestation is the process of vouching for the accuracy of information, such that a relying party can use the attestation to decide whether it trusts the remote platform. A platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a platform. Obviously, all forms of attestation require reliable evidence of the attesting element.
3. *Integrity Measurement and Storage*: Integrity measurement is the process of obtaining metrics of platform characteristics that affect the integrity (trustworthiness) of a platform; storing those metrics; and putting digests of those metrics in PCRs. An intermediate step between integrity measurement and integrity reporting is *integrity storage*. Integrity storage stores integrity metrics in a log and stores a digest of those metrics in PCRs.
4. *Integrity Reporting*: Integrity reporting is the process of attesting to integrity measurements recorded in a PCR. The report is signed using the private key (e.g. Attestation Identity Key (AIK) or restricted signing key) located in shielded locations in the TPM. Integrity measurement is a trusted function which "measures" (e.g. computes the hash) of components of the platform that are measurable (e.g. software, configurations etc.). The result is placed into an integrity measurement log. A digest of the measurements is then recorded in PCRs in the TPM so any tampering with the log can be detected. Reporting involves sending portions of the integrity measurement log to other parties (e.g. IMV) along with a signed set of PCRs which the other party can use to validate the logs contents prior to making trust decisions.

In addition to the fundamental features of trusted platforms that are mentioned above, in the context of Platform-Authentication (see TCG Infrastructure Working Group's Reference Architecture specification [4]), there are additional benefits that the TCG approach can provide:

- *Evaluation and Decision Making*: Following the TCG authentication model in [38], when a requestor platform issues a request (e.g. to resources) to a relying party, that relying party needs to make a trust decision about the requesting system's platform. The TCG model allows the relying party to evaluate the integrity measurements discussed above during this decision. Some relying parties may wish to delegate this evaluation to a 3rd party and merely review the results when making the decision. The outcome of platform evaluation is not limited to binary results (such as success/fail), but may include ranges of values (e.g. 1 to 100) indicating the level confidence the evaluating platform has with regards to its assessment.
- *Enforcement and Response*: Depending on the exact configuration of an evaluating platform, the platform may in fact be a Policy Enforcement Point (PEP) for a given set of environmental-specific policies. In addition, the platform may return *responses* to another platform, of whom it evaluated.

These features play an important role when an Endpoint seeks to obtain network access by reporting its integrity measurements to the PDP, which perform evaluation and decision-making regarding the access request, and which directs its evaluation results to the PEP for enforcement.

A TNC Client implementation makes use of the TPM and its functionality via a separate layer of services called the Platform Trust Services. (See Figure 5.) This layer provides some level of abstraction in order for both the TNC Client and the IMC to query their underlying platform trust information within the Endpoint on which they operate.

8.3 Roles

The roles in TNC Architecture do not change with the introduction of trusted platforms. However, the concept of platform ownership and the *owner* role should be considered. Among the roles identified by the TNC architecture are the Endpoint, PEP, and PDP. In some cases, the PEP and PDP have the same *owner*. In other words, they are controlled by the same IT department or service provider. The Endpoint may also have the same owner as PEP and PDP, but ownership should be re-validated before extending special privileges. Usually this is part of Platform-Authentication with a PDP.

In the case where the Endpoint and PDP owners are different, Platform-Authentication and remote attestation requires both parties to trust an Attestation Certificate Authority (ACA) who issues AIK-certificates to trusted platforms. In particular, the ACA is the element that seeks the Endpoint's EK-certificate and in-turn issues an AIK-certificate for the Endpoint's trusted platform. As such, when the Endpoint uses the AIK-certificate within a Platform-Authentication event, the PDP IMV needs to trust the same ACA and accept the AIK-certificate issued by that ACA. The components, protocols and interfaces described below support interactions between these elements.

8.4 Functions

In addition to previously described TNC functions, the TNC architecture includes additional functions when a TCG trusted platform makes up the host environment. The additional functions are described here:

- *Platform Trust Service* (PTS): The PTS is a system service that exposes trusted platform capabilities to TNC components. PTS services include protected key storage, asymmetric cryptography, random numbers, platform identity, platform configuration reporting and integrity state tracking.

- *The TPM Software Stack* (TSS): The TSS [39] is a middleware stack that enables applications to use higher level interfaces for communication with the TPM support functions. These include unlimited key storage (off-chip protected), key caching and higher-level interface abstraction.

- *The Trusted Platform Module* (TPM): The TPM hardware component implements protected capabilities, shielded locations, and other functions as described in [37].

8.4.1 Platform Trust Services

PTS architecture can be divided into four classes of functionality, namely TNC component integrity services, Platform-Authentication, trust transitivity and support for cryptography. The PTS may possess TPM *owner authorization* privileges as required to perform TPM operations. Some of these functionalities are described below, while others have been described in-depth elsewhere (see [4] and [37]).

8.4.1.1 TNC Component Integrity Services

The PTS provides measurement logs and ensures the logs accurately reflect the Platform Configuration Register (PCR) state. In addition to pre-boot and OS integrity state, the PTS can capture application integrity state.

The PTS exposes interfaces for Integrity Measurement Collectors (IMC) software to extend PCRs and write to integrity measurement logs. The PTS converts platform specific integrity log entries

into an interoperable format according to TCG integrity schema specifications. All log entries must be in the independent format before being sent over an IF-M interface. Therefore, TNC components should use the PTS for reporting entries in the Integrity Management Log.

The PTS provides access to TPM finite resources including key storage, content of PCRs, measurement logs, and transport sessions. It ensures processes and threads vying for access to these resources are serialized through appropriate process and thread locking mechanisms. Updates to the Integrity Measurement Log (IML) files are controlled such that log entries are synchronized with respect to PCR contents. An abstract representation of PCRs is exposed over IF-PTS to processes seeking to record and report integrity values.

8.4.1.2 Application Protocols

The PTS participates in protocols that establish verifiable platform identities, Platform-Authentication, and reporting of platform configuration state. The PTS is designed in such a way that it can be suitably deployed with tunneling protocols (e.g. within EAP), making use of Attestation Identity Keys (AIKs). The PTS may possess privileges necessary to use AIKs to perform other TPM protected operations.

Several protocols are anticipated to be supported by the TNC:

- Platform-Authentication using an AIK.
- Platform attestation using TPM PCRs and Integrity Measurement Log entries.
- Platform identity registration of AIK using the TPM EK.
- Platform monitoring protocol for reporting the presence of the platform integrity agents.

Other application protocols may be supported as determined by TNC requirements.

8.4.1.3 Trust Transitivity

The PTS provides component loading and registration services that can be used to capture integrity state of TNC components before execution threads are passed. The PTS cooperates with platform trust capabilities, including the Root of Trust for Measurement (RTM), to establish transitive trust linkages.

The PTS may employ any available platform specific anti-spoofing and anti-tampering techniques as necessary to strengthen trust assurances.

8.4.1.4 Security Considerations for Network Connection with TPM

Use of a TPM helps address a man-in-the-middle threat to the TNC Client and other components. TPM protected keys may be used to establish connections to PDP and PEP endpoints. Fixing the communications endpoint to hardware minimizes certain classes of MITM attacks (where a local redirector is involved).

The TPM platform configuration registers can be used to more reliably capture and report platform configuration information thereby reducing the threat of rogue software on the client platform performing MITM redirection.

The use of the TPM PCRs to validate the integrity measurement log prevents a system from lying about what the platform is running, so others can determine if the Endpoint has the desirable integrity. To close the vulnerability gap between the TPM and TNC components, a number of platform-specific techniques may be employed. While it is not the goal of the TNC architecture to define specific techniques, it is an objective to define interfaces for TNC components to be integrity checked prior to their being relied upon by Policy Decision Points, e.g. by having the TNC component measurements recorded in the PCRs.

8.5 Interface IF-PTS

PTS services and functionality is exposed to host processes through IF-PTS. Any of the TNC components may access PTS services through IF-PTS.

As a system service, the PTS must be discovered and the form of inter-process communication (IPC) established. Since a TNC component is an element in a transitive trust chain, mechanisms for measuring it and for transferring execution control must be established.

As an arbiter of finite resources, the PTS must have a way to publish available resources and a way to block access to allocated resources.

The operating status and error condition of the PTS must be available to subscribers. The PTS may start and stop while subscribers remain operational. Individual service requests should be acknowledged by success or failure notifications. In case of no acknowledgement, a timeout or keep-alive mechanism should be employed to ensure deterministic interaction semantics.

8.6 TNC and the TCG Integrity Management Model

The current TNC architecture accommodates platforms that possess a TPM and makes extensive use of the TPM as the hardware root of trust. Among others, this allows a PDP to gain some assurance that information regarding the Endpoint platform-state reported to by the PTS (on the Endpoint) is rooted in trust that is based on cryptographic information that is bound to the TPM hardware.

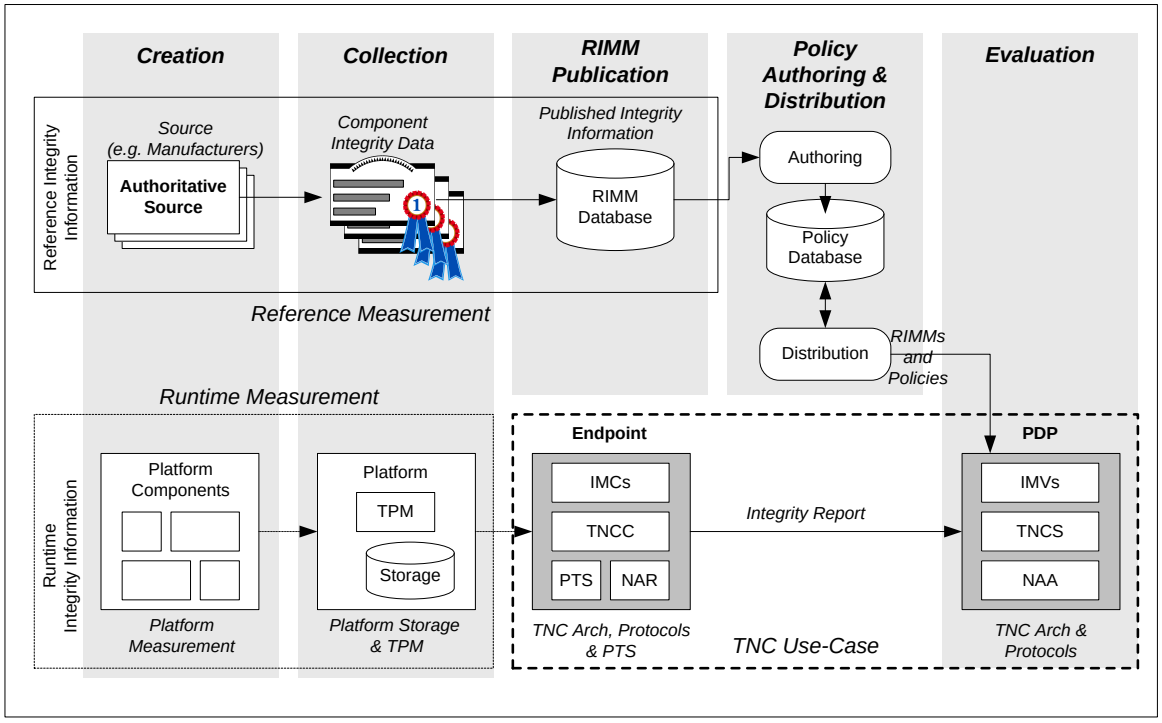

Figure 6: The TNC Architecture within the TCG Integrity Management Model

Although the TPM hardware itself provides a strong anchor of trust, another important dimension of trusted computing concerns the platform-state information that is being reported by the PTS in the Endpoint to the PDP. That is, there is the aspect of *how* the platform-state information is being reported to (i.e. protocols, methods) and there is the aspect of *what* platform-state information is being reported.

To that extent the TCG has developed an *Integrity Management Model* (IMM). Among others the purpose of the IMM is to define the lifecycle of platform-related information (e.g. component manufacturer, model, etc.) and define how this information affects the levels of trust accorded to components within a platform and thus to the platform as a whole.

The relationship of the TNC architecture and protocols within the TCG Integrity Management Model (IMM) is shown in Figure 6. Here, integrity management consists of five broad phases that are divided across two kinds of activities. The first set of activities - Reference Measurement - refers to the collection of (static) integrity information and data pertaining to components that make-up a platform. These measurements are likely to come from the manufacturer or developer of the component so their customers can recognize a valid instance of the component at run-time. This set of activities can be considered to be "out-of-band" from the perspective of a given use-case, such as trusted network connections, since they occur prior to the actual usage scenario that makes use of the integrity information.

The second set of activities - Runtime Measurement - pertains to the actual use of the integrity information within a given Platform-Authentication event, in which the integrity of the components of the platform (e.g. Endpoint) are measured and stored inside the Integrity Measurement Log (IML) of the platform and later used within a Platform-Authentication exchange, namely the TNC Integrity Check Handshake.

The TNC architecture and protocols play a crucial role in IMM as it represents a Platform-Authentication use-case (in the context of network access control) which makes use or consumes the integrity information collected and processed by the various phases of the IMM. More specifically, in Figure 6 the result of the runtime measurement of the Endpoint platform is communicated to the PDP (as the Evaluator) as part of the network access request of the Endpoint. The specific term used in this case is *integrity report* which represents the set of component integrity information about an Endpoint which is communicated by the Endpoint to the PDP within a Platform-Authentication event.

The PDP itself uses the policies inside the Policy Database (see Figure 6) pertaining to the Endpoints part of its decision-making regarding the Endpoint. It is important to note that besides traditional information within the Policy Database (e.g. user ID, ACL, etc.) the Policy Database contains additional information pertaining to the components of the Endpoint platform. More specifically, the Policy Database contains *Reference Integrity Measurement Manifest* (RIMM) records which denote the expected (golden) reference value for each component of the Endpoint platform. Using the RIMM information, the PDP is thus able to compare the reported component integrity information (in the Integrity Report communicated from the Endpoint) against a good benchmark or reference value as found in the Policy Database.

The RIMM information represents the end-product of the Reference Measurement phase. Among others, the RIMM contains integrity information from the manufacturer or vendor of the component which is source-authentic and which has been canonicalized according the TCG Core Integrity Scheme standard. The evaluator of a RIMM records will thus be able to verify the creator of the RIMM.

9 Security Considerations

9.1 Requirements and Assumptions

There are a number of requirements and assumptions with regards to the interfaces and messages of the TNC Architecture in Figure 5 from the perspective of security and message transport:

R1 *Number of messages*: There is no limit to the number of messages exchanged between an IMC and an IMV within a given Platform-Authentication event. (Note, however, that in practice there is a limited time for completing the authentication, and thus IMV/IMC implementers are encouraged to minimize the data exchanged, and the number of roundtrips required to complete their assessment. Note also that certain transports may impose limits on the number of round trips that may be used. For instance, IF-TNCCS-SOH 1.0 only permits a single round trip.)

R2 *Endpoint integrity checking as part of endpoint authentication and authorization*: When verifying security compliance of endpoints, the TNC Architecture requires that integrity checking be supported either as part of (during) an overall authentication/authorization event (e.g. user authentication, AIK-certificate validation, etc.), or as a separate event after (following) other forms of authentication have been performed. This allows re-verification of integrity information to be done independent of other authentication events (e.g. periodic checking of AV-status every few minutes vs. user-authentication at network logon time).

R3 *Ability to share information*: The TNC Architecture will allow the TNC elements to share information observed on the network so it can be factored into various security decisions.

R4 *Common security metadata schema*: When coordinating between distributed network security components, the standard metadata communicated between the TNC components will be structured according to common TNC Metadata Scheme(s).

A1 *Protection and reliability of message transport*: Since the integrity measurements data communicated between a TNC Client and TNC Server and metadata communicated between TNC components are not self-protecting, it is assumed that an underlying mechanism will provide for the protection of the data as it is delivered.

A2 *Platform-Authentication invocation*: For an Integrity Check Handshake, the IMC will always initiate by sending the first message in an authentication dialog between the IMC and the IMV.

9.2 Architectural Security

The current architecture document encompasses:

- aspects of endpoint posture between an Endpoint and a Policy Server, containing a TNC Client and TNC Server respectively;
- an optional MAP, which can interface the TNC Server with Sensors, Flow Controllers, and other MAP Clients; and
- an optional CMDB, which can store and distribute endpoint posture information from a Policy Server and other CMDB Clients.

There are a number of security aspects pertaining to the architecture as a whole that need to be highlighted, as these are relevant to implementations that seek to be conforming to the architecture and achieving security at the highest levels. These aspects are discussed in the following:

- *Secure Channels between Endpoint and Policy Server*: In order to communicate posture values and parameters between the TNC Client and the TNC Server, a secure channel must be established for this exchange. One possible location to establish this secure

channel is between the NAR (at the Endpoint) and the NAA (at the PDP); another is between the TPTC (at the Endpoint) and the TPTS (at the CEP). This channel must be end-to-end in the sense that the NAE (if present) must not gain access to the contents of this secure channel. The exact implementation of this secure channel is dependent on the purpose of the channel and the network configuration. An example of this channel would be one established through a tunneled EAP protocol - such as Protected EAP (PEAP), Tunneled TLS (TTLS), or Tunneled EAP (TEAP) - in the context of the 802.1X configuration. Similarly, an IKEv2 Phase-1 SA could be used to negotiate a special Phase-2 SA that then protects the posture information transfer in the case of a VPN.

- *Secure Channels between MAP and MAP Clients, and CMDB and CMDB Clients*: Metadata communicated between MAP Clients and the MAP, and endpoint information communicated between CMDB Clients and the CMDB, may be security sensitive. The confidentiality and integrity of this data must be preserved. Therefore, communication between these components must leverage secure transports (such as TLS).
- *Authorization for TNC Client/TNC Server and IMC/IMV*: In general, a TNCC/TNCS should only communicate with authorized IMCs/IMVs. This requirement comes from the need to prevent bogus IMCs/IMVs from opening communications with valid TNC Clients, thereby opening the possibility of a Denial-of-Service attack (at the very least) against the TNCC/TNCS.
- *Authorization for MAP and MAP Clients, and CMDB and CMDB Clients*: In general, a MAP Client should only communicate with authorized MAPs, and a CMDB Client should only communicate with authorized CMDBs. This requirement comes from the need to prevent distribution of information to or by unauthorized MAPs or CMDBs. In addition, a MAP should only communicate with authorized MAP Clients, and a CMDB should only communicate with authorized CMDB Clients, to prevent denial-of-service attacks from malicious clients and to protect the information store from corruption by unauthorized contributors. MAP Content Authorization should be used to control communications between MAP Clients and MAP Servers, and equivalent controls should be used between CMDB Clients and CMDBs.
- *Self-integrity of Endpoint and Policy Server*: The Endpoint and Policy Server must be protected against attacks that make unauthorized modifications to their system and platform configurations, because the posture values being communicated between two endpoints are only as good as the self-integrity of these entities. This need is particularly acute in the case of platforms without a hardware root of trust, such as a TPM, since such attacks would be harder to detect on such platforms.
- *Self-integrity of MAP and MAP Clients, and CMDB and CMDB Clients*: Compromise of the MAP, MAP Clients, CMDB, or CMDB Clients could lead to corruption of the MAP or CMDB databases. This could lead to undesired outcomes, such as network access being denied or allowed improperly, or distribution of misinformation to other network and security services. Therefore, the self-integrity of these components must be protected. These entities should also be protected against attacks and checked to ensure their ongoing health (e.g. using TNC posture checks, optionally also leveraging a TPM).
- *Security of Remediation Solutions*: In the event that remediation of an Endpoint requires that Endpoint to communicate with a remediation server and obtain posture-related updates, it is important to consider the security of the remediation server. If signed updates with careful versioning are placed on the remediation server, some protection against remediation server compromise can be achieved. However, strong protection for the remediation server should be employed.
- *Protection of Information Assets across Interfaces*: It is important that implementations of the TNC Architecture protect information assets as these traverse the various interfaces defined in the architecture. These information assets include, but are not limited to, state change notifications (between TNCC and IMV, and between TNCS and IMC), message

exchanges between elements, vendor specific messages (exchanged between the IMC and IMV as peers) and remediation results (from TNCC to the IMV).

- *Protection of Intra-Platform Component Discovery Mechanisms*: The ability of a TNCC on a platform to discover the IMCs on that platform has benefits as well as security risks. Thus, a TNCC must have sufficient privileges (set by the Administrator according to policy) to access information regarding available IMCs on the same platform. The design and implementation of interfaces must therefore protect against spoofing (by a rogue IMC/IMV), against denial of service provided by a legitimate IMC/IMV, and against unauthorized tampering (IMC/IMV parameters modified).

This section is only a brief summary of security considerations related to the TNC architecture. Each TNC interface specification includes an in-depth Security Considerations section that analyzes the security issues relevant to that interface and makes recommendations for appropriate countermeasures. Each interface specification also contains normative requirements for countermeasures relevant to that interface. All parties are urged to review these sections in detail to understand and properly implement these countermeasures.

10 Privacy Considerations

Privacy is an important issue in the context of trusted network communications. Some aspects that are pertinent to the TNC are as follows:

- *Anonymous access is supported*: User authentication (of the client system to the server system) is not required in order to perform an integrity measurement handshake. In scenarios which require protection of the user identity, anonymous network access is supported by this architecture.
- *Owner controlled policy*: The architecture allows for negotiation of which measurements may be needed to make access decisions. The platform owner is presumed to have control over the privacy policy and privacy related negotiations. In other words, measurements can be more specific than what is requested and client policies can dictate when it is desirable to abort the connection request in the interest of preserving privacy.
- *Disclosure control mechanisms*. The architecture does not prevent IMCs from implementing a disclosure control mechanism driven by privacy policy. IMC implementers may employ filtering on outbound flows to block, replace, modify or un-sign posture reports. The IMC interface specification does not specify the content of messages exchanged between IMC and IMV; hence the TNCC does not appear to be an appropriate place to apply privacy controls. However, vendor specific extensions to IMCs appear reasonable.
- *IMC selection*. The user may determine which IMCs can be installed and/or loaded by the TNCC based on an assessment of the IMC ability to protect privacy.
- *Protection of sensitive information*: Measurement information provided by a client needs to be protected once it leaves the client, in order to ensure confidentiality of the sensitive measurement data (such as PII) and prevent disclosure to unauthorized parties. This includes protecting the information in transit as well as in the MAP and/or CMDB through mechanisms such as encryption and controlled, authorized access.
- *Anonymity of published information:* MAP and CMDB Clients may publish information such as endpoint posture, network access, events (which may include information about what services an endpoint is accessing), roles and capabilities, and the identity of the end user operating the endpoint. Any of this published information may be queried by other MAP or CMDB Clients and could potentially be used to correlate network activity to a particular end user. Care should be taken by deployers of these components to ensure that the information published by MAP or CMDB Clients does not violate agreements with end users or local laws and regulations.

These measures help ensure that privacy can be properly protected in the TNC architecture.

11 References

[1] P. Sangster et al., *Network Endpoint Assessment (NEA): Overview and Requirements*, RFC 5209, IETF Informational, June 2008.

[2] Trusted Computing Group, *TCG Trusted Platform Module (TPM) Specifications v1.2*, March 2011.

[3] Trusted Computing Group, *TCG Trusted Platform Module (TPM) Specifications v2.0*, March 2011.

[4] Trusted Computing Group, *IWG Reference Architecture for Interoperability (Part 1)*, Specification Version 1.0, June 2005.

[5] P. Sangster, K. Narayan, *PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)*, RFC 5792, IETF Standards Track, March 2010.

[6] R. Sahita, S. Hanna, R. Hurst, K. Narayan, *PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC)*, RFC 5793, IETF Standards Track, March 2010.

[7] P. Sangster, N. Cam-Winget, J. Salowey, *A Posture Transport Protocol over TLS (PT-TLS)*, RFC 6876, IETF Standards Track, February 2013.

[8] N. Cam-Winget, P. Sangster, *PT-EAP: Posture Transport (PT) Protocol for Extensible Authentication Protocol (EAP) Tunnel Methods*, RFC 7171, IETF Standards Track, May 2014.

[9] The International Organization for Standardization/International Electrotechnical Commission, *Information Technology – Software Asset Management - Part 2: Software Identification Tag*, ISO/IEC 19770-2, November 2009

[10] Trusted Computing Group, *TNC IF-M: TLV Binding v1.0*, May 2014.

[11] Trusted Computing Group, *TNC IF-TNCCS: TLV Binding v2.0*, May 2014.

[12] Trusted Computing Group, *TNC IF-T Binding to TLS v2.0*, February 2013.

[13] Trusted Computing Group, *TNC IF-T: Protocol Bindings for Tunneled EAP Methods Specification 2.0*, May 2014.

[14] Trusted Computing Group, *TNC IF-IMC Specification v1.3*, February 2013.

[15] Trusted Computing Group, *TNC IF-IMV Specification v1.4*, December 2014.

[16] Trusted Computing Group, *Attestation PTS Protocol: Binding to IF-M*, August 2011.

[17] Trusted Computing Group, *TNC SWID Messages and Attributes for IF-M Specification v1.0*, August 2015.

[18] Trusted Computing Group, *TNC IF-M Segmentation v1.0,* April 2016.

[19] Trusted Computing Group, *TNC IF-TNCCS Specifications v2.0*, May 2014.

[20] Trusted Computing Group, *TNC IF-TNCCS-SOH Specification v1.0*, May 2007.

[21] Trusted Computing Group, *IWG IF-PTS Specification v1.0*, November 2006

[22] Trusted Computing Group, *IWG Simple Object Schema Specification v1.0,* November 2006.

[23] Trusted Computing Group, *IWG Core Integrity Schema Specification v2.0,* August 2011.

[24] Trusted Computing Group, *IWG Integrity Report Schema Specification v2.0,* August 2011.

[25] Trusted Computing Group, *IWG Reference Manifest (RM) Schema Specification v2.0,* August 2011.

[26] Trusted Computing Group, *IWG Security Qualities Schema Specification v1.1,* May 2007.

[27] Trusted Computing Group, *IWG Verification Result Schema Specification v1.0,* May 2007.

[28] Trusted Computing Group, *TNC IF-PEP: Protocol Bindings for RADIUS Specification v1.1*, February 2007.

[29] Trusted Computing Group, *TNC IF-MAP Binding for SOAP Specification v2.2*, March 2014.

[30] Trusted Computing Group, *TNC IF-MAP Metadata for Network Security v1.1*, May 2012.

[31] Trusted Computing Group, *TNC IF-MAP Metadata for ICS Security v1.0*, September 2014.

[32] Trusted Computing Group, *TNC MAP Content Authorization v1.0*, June 2014.

[33] Trusted Computing Group, *TNC Endpoint Compliance Profile v1.0*, December 2014.

[34] Trusted Computing Group, *TNC CESP Specification v1.0,* May 2009.

[35] Trusted Computing Group, *TNC Federated TNC Specification v1.0*, May 2009.

[36] Trusted Computing Group, *TNC Server Discovery and Validation v1.0*, October, 2017.

[37] Trusted Computing Group, *Trusted Platform Module Library Part 1: Architecture TPM 2.0 L00 R01.16*, October 2014.

[38] Trusted Computing Group, *TCG Architecture Overview*, Revision 1.4, August 2007.

[39] Trusted Computing Group, *TCG Trusted Software Stack (TSS) Specifications v1.2*, January 2006.

[40] Trusted Computing Group, *TCG Glossary*. See https://www.trustedcomputinggroup.org/glossary/

[41] IEEE802, *Port-Based Network Access Control*, IEEE Std 802.1X-2004, December 2004, Institute for Electrical and Electronics Engineers (IEEE).

[42] P. Congdon, B. Aboba, A. Smith, G. Zorn, J. Roese, *IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*, RFC 3580, IETF Informational, September 2003.

[43] C. Rigney, S. Willens, A. Rubens, W. Simpson, *Remote Authentication Dial In User Service* (RADIUS), RFC 2865, IETF Standards Track, June 2000.

12 **TNC Glossary**

When used in TNC documents, the following terms are defined as below. Please also see the TCG Glossary [40] for other TCG-related terminology.

Term	**Definition**
Access Control Capability	The TNC capability that decides which Endpoint actions should be permitted - based on information consumed from the Compliance capability, the Orchestration capability, and/or locally configured access policies - and enforces these decisions.
Access Control Enforcer (ACE)	The function of a MAP Client that aids in on-going and granular enforcement of network security policy compliance based on information consumed via IF-MAP. The Access Control Enforcer might not be directly involved with initial network access decisions nor directly connected to an Endpoint.
Administrative Client	The function of a MAP Client that enables administrative operations such as monitoring, investigation, and provisioning. The Administrative Client might not be directly involved with initial network access decisions nor directly connected to an Endpoint.
Back-Haul Interface (BHI)	The function of a MAP Client that segregates protected ICS network devices communicating across a shared, untrusted commodity IP infrastructure. The BHI implements necessary authentication, encryption, translation, and authorization policy enforcement capabilities to create the overlay network, using IF-MAP for coordination, provisioning, and management.
Configuration Management Database (CMDB)	The role of an element in the TNC framework that stores collected endpoint measurements.
CMDB Client	The role of an element in the TNC framework that communicates endpoint information to and consumes it from CMDBs.
Compliance Capability	The TNC capability that enables an administrator to collect compliance reports from Endpoints and evaluate these reports against network policy to identify non-compliant Endpoints.
Compliance Evaluation Point (CEP)	The role of an element in the TNC framework that collects and evaluates endpoint posture information, and sends the information to a CMDB for storage.
Endpoint	The role of an element in the TNC framework seeking connectivity to, or already connected to, a network that implements the TNC Architecture.
Endpoint Compliance	An indication of whether or not an Endpoint's posture meets the requirements of its environment. Determining compliance requires establishing a level of 'trust' in the state of an Endpoint by verifying aspects such as the presence, status, and upgrade level of mandated applications; revisions of signature libraries for anti-virus and intrusion detection and prevention system applications; the patch level of the Endpoint's operating system and applications; and other posture measurements.
IMV Action	The recommendation given by each IMV to the TNCS as to what type

Recommendation	of network access or isolation action should be taken based on the IMV's evaluation. Example IMV Action Recommendations include: recommend full (normal) access; recommend isolation (limited or quarantined access); and recommend denial (no access).
IMV Evaluation Result	The result returned by each IMV to the TNCS regarding the state of the Endpoint's compliance, based on the IMV's evaluation. Example IMV Evaluation Results include: Endpoint is compliant with policy; Endpoint is non-compliant and non-compliance is minor; Endpoint is non-compliant and non-compliance is major; compliance is unknown.
Integrity Check Handshake	The handshake between a TNCC and a TNCS during which the posture of an Endpoint is checked against policy to determine whether the Endpoint should be given access to a resource. An Integrity Check Handshake is a TNC-related instance of a TCG attestation protocol (see [38]).
Integrity Information	The set of platform specific information that makes up a Trusted Platform. This ranges from information about a platform's hardware components, TPM information (e.g. versions), PCRs, peripherals, Trusted Building Blocks, OS/Kernel, drivers, Applications, Anti-Virus information and others. Each specific use-case determines which information set will be of interest. As such, it is expected that for a given situation these will be pre-determined or pre-configured by an authorized entity (e.g. IT administrator).
Integrity Measurements	TNC measurements provided by IMCs that communicate Endpoint integrity information.
Integrity Measurement Collector (IMC)	The collection/verification layer function of an Endpoint that measures certain aspects of the Endpoint's posture, including software versions, patches, Anti-Virus and others. An IMC may use the TCG Platform Trust Service (PTS) to obtain integrity information regarding every component of the platform on which the IMC sits. Multiple IMCs may reside on a single Endpoint.
Integrity Measurement Verifier (IMV)	The collection/verification layer function of a Policy Server that verifies a particular aspect of the Endpoint's posture, based on measurements received from an IMC and/or other data. Multiple IMVs may reside on a single PDP.
Isolation	The action of separating an Endpoint onto a separate network - virtual or physical - possibly, though not necessarily, for the purposes of performing Remediation on that Endpoint.
MAP Service	One or more MAPs providing publish/search/subscribe access to a single MAP Graph representing the state of a given network.
Metadata Access Point (MAP)	The role of an element in the TNC framework that serves as a broker/server to which metadata may be published, and from which metadata may be searched and subscribed to, using the IF-MAP protocol.
Metadata Access Point Client (MAPC)	The role of an element in the TNC framework that publishes metadata to or searches / subscribes to metadata from a MAP.
Metadata Access Point Server (MAPS)	The function of a MAP that allows MAP Clients to publish, subscribe to, and search metadata.
Network Access	The decision sent from an NAA to an NAE via IF-PEP to control an Endpoint's network access. This decision may be a simple binary

Decision	value (allow or deny network access) or it may include information (such as a VLAN ID) for purposes such as Isolation. Alternatively, it may include information (such as a VLAN ID) for purposes such as Isolation.
Network Access Authority (NAA)	The network access layer function of a PDP that decides whether a Network Access Requestor (NAR) should be granted access to a network.
Network Access Enforcer (NAE)	The network access layer function of a PEP that consumes and enforces access control policies from a Network Access Authority (NAA).
Network Access Requestor (NAR)	The network access layer function of an Endpoint responsible for negotiating and establishing network access onto a given network. The NAR is expected to implement network layer protocols, covering security, message transport and others. In the context of 802.1X, the NAR can be identified as the Supplicant.
Orchestration Capability	The TNC capability that offers a notification service and unified, extensible data model, enabling network and security devices to better perform their functions, and to share context with the Orchestration capability to enable other elements to better perform their functions.
Platform Authentication	The act of verifying both the proof-of-identity and integrity-status of a given platform.
Platform Trust Services (PTS)	A system service that exposes trusted platform capabilities to TNC components that reside on a Trusted Platform containing a TPM. PTS services include protected key storage, asymmetric cryptography, random numbers, platform identity, platform configuration reporting and integrity state tracking.
Policy Decision Point (PDP)	The role of an element in the TNC framework that evaluates the status of a TNC Client (seeking network connectivity) and decides upon some network-related action to be enforced by a PEP. The PDP embodies the security and compliance related policies governing the network.
Policy Enforcement Point (PEP)	The role of an element in the TNC framework that controls access to a protected network, whose policies are implemented through a Policy Decision Point (PDP). The PEP enforces the decision of the PDP.
Policy Server	A server that applies policy to some set of actions. In the TNC Architecture, both the PDP and the CEP are Policy Servers; a NEA Server is also a Policy Server. The Policy Server collects and evaluates endpoint posture information (CEP, NEA Server) and/or makes access control decisions based on endpoint context (including role, compliance, location, behavior, and other factors) and communicates those decisions to enforcement points (PDP).
Posture Measurements	TNC measurements provided by IMCs that communicate Endpoint posture information including endpoint provisioning, installed software, and/or configuration.
Remediation	The action of updating an element seeking network connectivity (that fails posture check) with the necessary software, firmware and posture-related parameters updates.

Resource	Network, service, application, and/or information accessed by an endpoint.
Root of Trust	A component that is trusted always to behave in the expected manner, because its misbehavior cannot be detected. It performs one or more security-specific functions, such as measurement, storage, reporting, verification, and/or update. The root of trust provides an initial source of trust for an endpoint. A hardware TPM is an example of a hardware root of trust.
Sensor	The function of a MAP Client that monitors network and endpoint activity and then shares information with other TNC components through IF-MAP. The Sensor may not be directly involved with initial network access decisions nor directly connected to an Endpoint.
TNC Client (TNCC)	The evaluation layer function of an Endpoint that aggregates posture measurements (from IMCs), assists the management of the Integrity Check Handshakes, and assists in the measurement and reporting of platform and IMC integrity.
TNC Server (TNCS)	The evaluation layer function of a PDP that manages the flow of messages between Integrity Measurement Collectors (IMCs) and Integrity Measurement Verifiers (IMVs), gathers recommendations from IMVs, and combines those recommendations (based on policy) into an overall TNCS Action Recommendation to the NAA.
TNCS Action Recommendation	The final, combined recommendation given by the TNCS to the NAA. TNC specifications do not currently mandate values for this recommendation; however, example action recommendations are expected to include: recommend full (normal) access; recommend isolation (limited or quarantined access); and recommend denial (no access).
TNC Posture Transport Client (TPTC)	The network access layer function of an Endpoint that facilitates network communication with a TPTS, over which the TNCC communicates posture information.
TNC Posture Transport Server (TPTS)	The network access layer function of a Policy Server that receives endpoint posture information from a TPTC.
Trusted Platform Module (TPM)	A cryptographic processor that implements the functions defined in the TCG Trusted Platform Module Specification; the set of Roots of Trust with shielded locations and protected capabilities.

13 Appendix A: User Communities

This document is written with two broad classes of users in mind: users of the document itself, whose goal may be to develop products, design networks, and/or draft other standards based on the TNC Architecture, and users of the technology enabled by the TNC Architecture. Users of TNC-enabled technology further subdivide into primary users, who are designing, deploying, and maintaining secure environments, and secondary users, who interact with and consume data from such environments.

13.1 Users of This Document

User Role:	**Product Implementer**
Primary Goal:	Increase customer value by developing a product that interoperates with other network devices
Background:	Technical- and business-minded. Has a product (or an idea for a product) that requires interoperability with other network devices. Does not want to reinvent the wheel for basic network security functions. Needs an introduction to the "big picture" of what TNC offers.
Typical Usage:	Reads to understand how TNC standards all fit together, and to consider business cases in which their ideas converge with the security TNC offers. Wants to be directed to specific requirements for various interfaces and protocols that fit their business case.
Motivations and Expectations:	Often profit-driven. Needs clear requirements and interoperability testing to ensure that their product will work when it goes to market.

User Role:	**Solution Architect**
Primary Goal:	Protect their environment by designing a network that is resistant to outsider attacks and insider threats.
Background:	Primary consideration is making sure the network works for the user; but, security is a large consideration. Wants robust security, but has difficulty making many network security products work together. Cannot allow security to hinder the work of the network user.
Typical Usage:	Reads to understand the benefits of TNC for security and interoperability. Wants to understand what to ask for when speaking to vendors. Needs clarity in use cases to enable vision of how TNC products could solve pressing network security products.
Motivations and Expectations:	Interoperability, ease-of-use, and security. Needs big picture, not detailed product requirements.

User Role:	**Specifications and Standards Developers**
Primary Goal:	Write standards or specifications for interoperability and security; promote adoption of standards
Background:	Deeply technical; understands how everything fits together from the bottom up; interested in technical architecture.

Typical Usage:	Leverage TNC framework to support other specifications and standards efforts, and to ensure compatibility with existing TNC standards.
Motivations and Expectations:	Use existing body of work, rather than reinventing the wheel, for trusted network communications.

13.2 Primary Users of TNC-Enabled Technology

User Role:	**Solution Architect**
Primary Goal:	Protect their environment by designing a network that is resistant to outsider attacks and insider threats.
Background:	Primary consideration is making sure the network works for the user; but, security is a large consideration. Wants robust security, but has difficulty making many network security products work together. Cannot allow security to hinder the work of the network user.
Typical Usage:	Looking for TNC standards-based products that are interoperable and can be expected to solve one or more clearly defined problems. Needs these solutions to fit in to existing network.
Motivations and Expectations:	Interoperability, ease-of-use, and security. Needs big picture, not detailed product requirements.

User Role:	**Solution Implementer**
Primary Goal:	Reduce operational cost by setting up network equipment efficiently and easily.
Background:	Detailed technical knowledge of how the network works (and what changes would make it break). Provides valuable input during acquisition and architecture design.
Typical Usage:	Configures machines to accomplish network security goals. Leverages knowledge of standard to know what functionality is supported and how to configure devices to support security use cases.
Motivations and Expectations:	Wants to know what features are supported and how to configure them. Desires a common language for communicating between devices on the network to simplify job. Needs extensive understanding of the details of the protocols and schema each machine uses.

User Role:	**Systems Administrator (Operations and Maintenance)**
Primary Goal:	Increase efficiency by keeping network resources available to authorized users and secure from unauthorized users.
Background:	Often juggling multiple projects and operating with limited resources. . Using many non-interoperable tools to manage network oversight. Must balance security needs with user needs. Has to prove compliance to regulations, but often lacks the ability to gather necessary data.

Typical Usage:	24/7 response to threats, vulnerabilities, access requests, etc.
Motivations and Expectations:	Wants network security solutions that work together, and that do not require a lot of time to manage. Wants to be able to prevent attacks while not denying access to users who are authorized. Wants to be able to "check the boxes" for regulation without expending a lot of energy.

13.3 Secondary Users of TNC-Enabled Technology

User Role:	**Enterprise Endpoint User**
Primary Goal:	Access network resources
Background:	Has a job to perform and needs to access network resources to do it. Is willing to work around security as needed to get job done. Does not see the security of the network as their primary mission.
Typical Usage:	Day-to-day access to resources, both on enterprise owned and personally owned devices.
Motivations and Expectations:	Wants devices and network access to "just work". Does not want to spend any time at all on security. Sees security measures as a hindrance to productivity.

User Role:	**Regulators**
Primary Goal:	Ensure network's compliance to federal, industrial or organizational regulation
Background:	Not primarily a consumer of network resources. Sees security of the network as primary concern.
Typical Usage:	Generates periodic evidence of network compliance, in the form of reports and test results.
Motivations and Expectations:	Wants complete insight into network state and health. Wants alerts when network monitoring tools detect non-compliance. Requires consistent means of expressing network security data to enable compliance report generation.

14 Appendix B: Relation to TCG IWG Architecture

The TNC Architecture is derived from the broader IWG Architecture [4]. Therefore, the Platform-Authentication model underlying the IWG Architecture also underlies the TNC Architecture. This is shown in Figure 7, with mappings to the TNC Architecture.

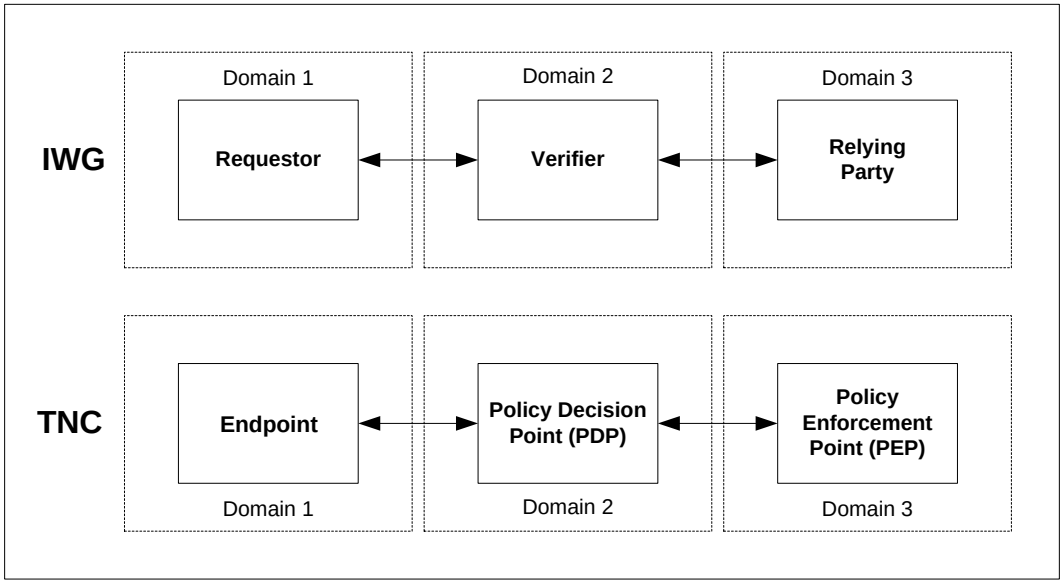

Figure 7: Basic Model underlying the IWG and TNC Architectures

In the IWG architecture when responding to a request from a *Requestor* element, a *Relying Party* is dependent on the decision outcome of a *Verifier*. This basic behavior maps quite readily to the basic network connection request behavior, in which a network capable device (e.g. a client or 802.1X Supplicant) seeks network connectivity and access to resources available on the network, through another device (e.g. 802.1X Authenticator, switch) relying on the permissions decision of a third device (e.g. AAA Server) [41].

In the TNC architecture, the Endpoint acts as an IWG Requestor, the TNC PDP acts as an IWG Verifier, and the TNC PEP acts as an IWG Relying Party.

Though not visible in Figure 7, another important aspect shared between the two architectures is the use of the trusted computing feature of *integrity measurement* and *integrity verification* to establish a decision regarding a network access request. It is this hardware-rooted feature that distinguishes the IWG and TNC architectures from other architectures.

15 Appendix C: Assessment, Isolation, and Remediation

Although not visibly evident within the TNC Architecture of Figure 5, one important feature of the architecture is its extensibility and support for the isolation and remediation of Endpoints which do not succeed in obtaining network access permission due to failures in posture verification. Figure 8 shows an additional layer addressing remediation and provisioning.

Note that in the current TNC Architecture document, remediation is out of scope and is treated briefly for completeness.

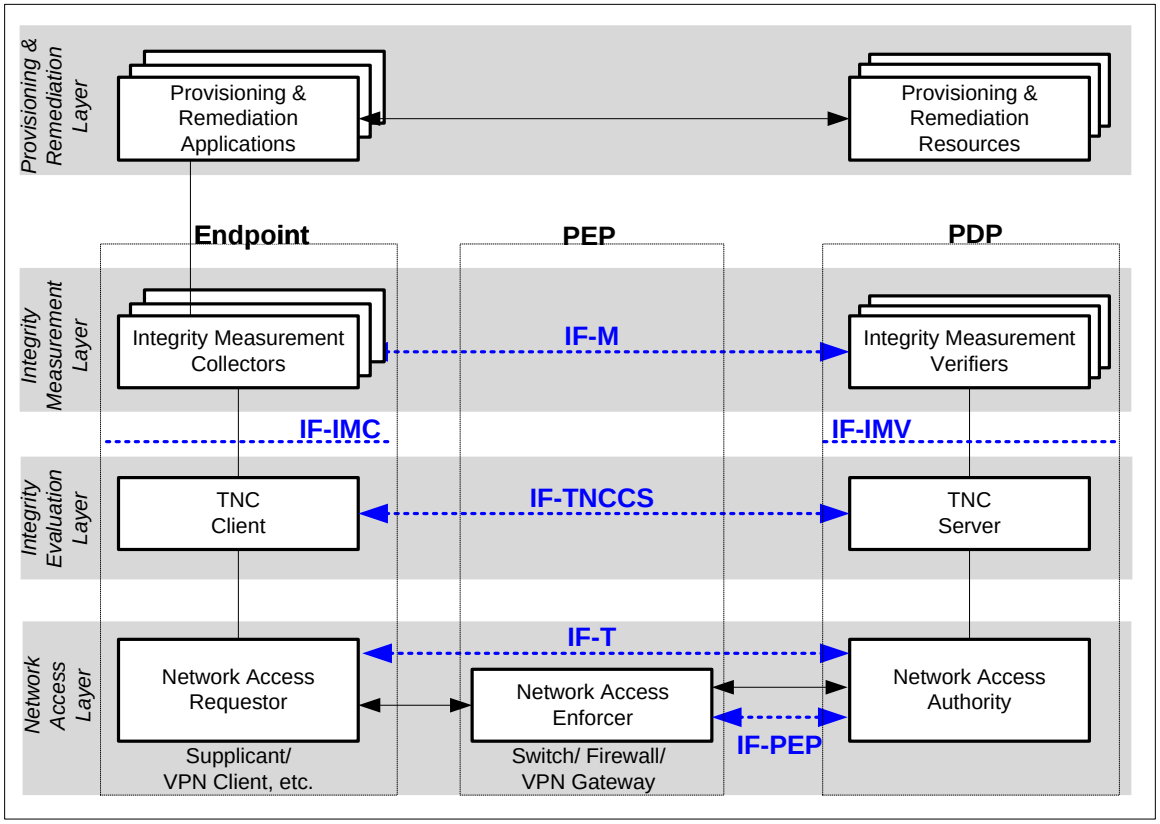

Figure 8: The Provisioning and Remediation Layer in the TNC Architecture

15.1 Phases in Network Access Control

In order to understand the actions needed to remedy Endpoints that fail posture verification, it is useful to view network connection requests in three basic phases from the perspective of posture verification:

- *Assessment*: In this phase, the IMVs perform the verification of the Endpoint following the policies set by the Network Administrator and if necessary delivers remediation instructions to the IMCs.
- *Isolation*: If the Endpoint has been authenticated and is recognized to be one that has some privileges on the network but has not passed the posture-verification by the IMV, the PDP may return instructions to the PEP to redirect the Endpoint to an isolation environment where the Endpoint can obtain posture-related updates.
- *Remediation*: Remediation is the process of the Endpoint obtaining corrections to its current platform configuration and other policy-specific parameters in order to bring it in line with the PDP's requirements for network-access of the PDP.

15.2 Assessment Phase

In the Assessment Phase, the TNC Client reports its current posture status to the TNC Server. Upon receiving the client posture status, the IMVs with the aid of the TNCS perform an assessment of the Endpoint based on the set of policies defined by the network administrator. The IMV can make one of three IMV Action-Recommendations (Allow, Isolate or Block) or it can make no recommendation.

If the platform is a Trusted Platform that deploys a TPM, then certain basic verifications, such as authenticating the platform's AIK-certificates, should be verified first before other more platform-specific verifications are performed.

At this point, it is important to note that the TNCS dialog with the TNCC may consist of several rounds of messages, where in each round the IMVs request more detail. This represents an extension to the basic behavior of the TNCC simply reporting all its posture information in a single set of messages to the TNCS.

If the IMVs find that remediation is needed, they will typically send remediation instructions to the IMCs in the final message of their dialog. The IMCs may execute these instructions immediately or hold them until some form of network access is available.

15.3 Isolation Phase

An important tool in the effort to remediate Endpoints that fail posture verification is the isolation of that Endpoint to a separate network - referred to here as the Isolation Network - in order to provide remediation services to the Endpoint. This protects the Endpoint from the full network and vice versa, preventing the spread of viruses and worms. There are a number of technical approaches today to achieve network isolation for the Endpoint. Two of these are as follows:

- *VLAN Containment*: VLAN containment permits the Endpoint to access the network in a limited fashion. Typically the primary purpose of the limited access is to allow the Endpoint to access on-line sources of remediation data (e.g. virus definition file updates, worm removal software, software patches, etc.). In some cases, no remediation is offered and the Endpoint is instead offered access to limited services, in such a fashion as to limit the potential for impact to the network or other attached hosts. RADIUS provisions VLAN containment using the Tunnel-Private-Group-ID attribute, as specified in RFC 3580 [42].
- *IP Filters*: In the case of IP filters, the PEP is configured with a set of filters which defines network locations reachable by the isolated Endpoint. Packets from the Endpoint destined to other network locations are simply discarded by the PEP. RADIUS selects filter rules for application to a network access session using the Filter-ID attribute (see RFC 2865 [43] and RFC 3580 [42]).

15.4 Remediation Phase

The TNC Architecture in Figure 8 accommodates a number of schemes for remediation. The intent of remediation is generally universal, namely that of performing updates to the software and firmware of the Endpoint to help it comply with the current network policy.

The general aim of remediation is to bring the Endpoint up to date in all posture-related information, as defined by the current policy for authorization. Examples include OS patches, AV updates, firmware upgrades, etc. Section 15.5 below discusses the TNC approach to remediation in further detail.

After remediation has been completed, the IMCs can ask the TNCC to retry the Integrity Check Handshake, which results in another Assessment Phase. This second phase may be shorter than the first since the IMCs may be able to send only the data that has changed (if supported by the IMVs).

15.5 Remediation in the TNC Architecture

The TNC Architecture supports remediation, both from the trusted network communications (endpoint posture) perspective, and from the broader TCG platform manageability perspective. In the Architecture, elements that take on a specific role may have additional functions in other contexts beyond endpoint posture.

The TNC Architecture support for remediation and provisioning is expressed in the corresponding *Provisioning & Remediation layer* in Figure 8. The layer contains applications, services and other resources necessary to establish and maintain a trusted platform according to the platform owner's specifications. It is relevant not only for the remediation needs of trusted network communications - where enterprises can keep their system up to date - but also for the broader needs of Trusted Computing. These may include any of the following:

- Compliance and policy evaluation
- Collection / distribution of baseline measurements
- Provisioning of policies, settings, software and firmware
- Trusted-platform specific operations (see Section 8).

There are two elements relevant to remediation in the TNC Architecture (see Figure 8):

- *Provisioning & Remediation Applications (PRA)*: The Remediation Application can be implemented in several forms. For example, the PRA could be implemented as part of the Endpoint. Here, the PRA communicates with the IMC and provides it with specific types of posture information. An example of an embodiment of the PRA would be the Anti-Virus application software that communicates with sources of Anti-Virus parameters (e.g. latest AV signature files). Note that the PRA could be implemented as part of the IMC. As another example, the PRA/IMC could be an agent that updates the TPM and the TSS (part of the PTS), which obtains updates from the TPM Manufacturer.

Provisioning & Remediation Resources (PRR): The PRR represents the various sources of posture information needed to update the Endpoint so that it can be successfully verified by the PDP at the next re-attempt of the handshake. Examples of the PRR include enterprise servers, vendor services (e.g. FTP server), CDs/DVDs containing the update parameters, and others.

16 Appendix D: Basic Message Flows for Network Admission

There are a number of fundamental message types that are exchanged between components in the architecture, across the various interfaces defined above. The basic messages required for granting access to the network are summarized in [28] and are described in the following. Note that in the following illustration, several levels of authentication and authorization are assumed to have been configured to occur before a connection request can be completely fulfilled. In this example, these consist of the following order: User Authentication, Platform Credential Authentication and Integrity Check Handshake. Note, however, that in other situations this may not necessarily be the order of processing. Detailed examples of metadata-sharing messages which may augment the process below and provide other coordination between TNC components can be found in [29] [30] [31].

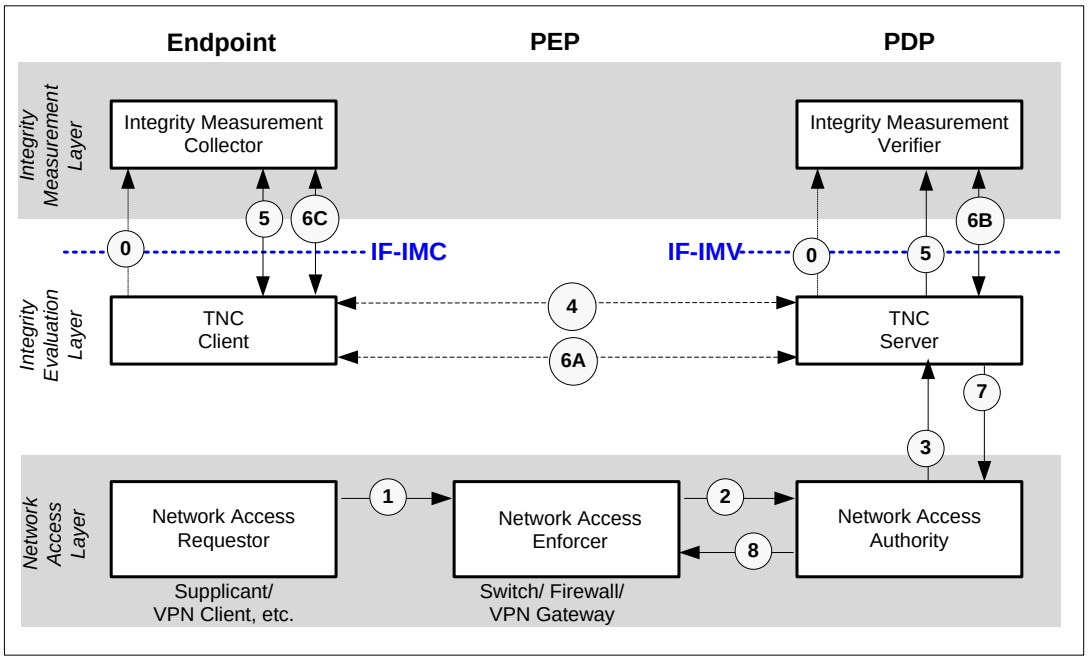

Figure 9: Message Flow between Access Granting Components in the TNC Architecture

- **Flow 0**: Prior to beginning a network connection and Integrity Check Handshake attempt, the TNCC must discover and load each relevant Collector using the platform-specific binding. The TNCC must then initialize the IMC, which includes defining the necessary connection IDs and Collector IDs, and ensuring that the TNCC has a valid connection state with the IMC.

  During the load process, the TNCC may check the integrity of the IMCs. This is optional. If a TPM is present, this check will typically involve hashing the IMCs and adding their hashes to a PCR (i.e. performing one or more TPM Extend operations). If no TPM is present, this check may involve checking the signatures on the IMCs. Integrity checks during IMC loading are done completely by the TNCC since there is no TNCS or IMV available. TNCS and IMVs will get a chance to do platform authentication of the Endpoint platform later in the sequence of events.

Similarly, the TNCS must discover and load each relevant IMV using the platform-specific binding.

- **Flow 1:** When a network connection attempt is triggered (automatically or by user request), the NAR at the Endpoint (client) initiates a connection request at the link and network layers.
- **Flow 2:** Upon receiving a network connection request (from the NAR), the NAE sends a network access decision request to the NAA. Here, the NAA is assumed to have been configured to perform User Authentication, Platform Credential Authentication and Integrity Check Handshake.

  User authentication can occur between the NAA and the NAR. Platform Credential Authentication and Integrity Check Handshake may have occurred between the Endpoint and the TNCS.

  Note that since an ordering of authentication has been configured, failure in one authentication will discontinue other forms of authentication and integrity check. That is, if the user fails user authentication with the NAA, then Platform Credential Authentication and Integrity Check Handshake will not proceed.
- **Flow 3:** Assuming that User Authentication succeeded between the user (on the Endpoint) and the NAA, the NAA then informs the TNCS of the connection request.
- **Flow 4:** The TNCS then performs (mutual) Platform Credential Authentication with the TNCC, verifying, for example, that valid (un-revoked) AIK-credentials are used by both entities.
- **Flow 5:** Assuming that Platform Credential Authentication succeeds between the TNCS and TNCC, the TNCS indicates to the IMVs (using interface IF-IMV) that a new connection request has occurred and that an Integrity Check Handshake needs to be carried out by the TNCS. Similarly, the TNCC indicates to the IMCs (using interface IF-IMC) that a new connection request has occurred and that an Integrity Check Handshake needs to be carried out by the TNCC. The IMCs respond by giving a number of IMC-IMV messages to TNCC across IF-IMC.
- **Flow 6A:** In order for an Integrity Check Handshake to occur, the TNCS and TNCC begin the exchange of messages pertaining to the integrity check. These messages will be relayed through the NAR, NAE and NAA, and will continue until the TNCS is satisfied with the integrity status of the Endpoint. Flow 6A shows this as a peer connection between the TNCS and TNCC.
- **Flow 6B:** The TNCS passes each IMC message to the matching IMV or IMVs through IF-IMV (using message types associated with the IMC messages to find the right IMV).

  Each IMV analyzes the IMC messages. If an IMV needs to exchange more messages (including remediation instructions) with an IMC, it provides a message to the TNCS through IF-IMV. If an IMV is ready to decide on an IMV Action Recommendation and IMV Evaluation Result, it gives these to the TNCS through IF-IMV.
- **Flow 6C:** Similarly, the TNCC will forward messages from the TNCS to the matching IMC or IMCs through IF-IMC, and send messages from the IMCs to the TNCS.
- **Flow 7:** When the TNCS has completed its Integrity Check Handshake with the TNCC, it then sends its TNCS Action Recommendation to the NAA. Note that the NAA may still have the option of not granting network access if other security

policy requirements have not been met by the Endpoint (even though the Endpoint has passed the Integrity Check).

- **Flow 8:** The NAA then sends its network access decision to the NAE to enforce. The NAA must also indicate its final decision to the TNCS which will be sent to the TNCC. Typically, the NAE indicates its execution of the decision (e.g. Port open in 802.1X) to the NAR.
- The above represents the basic behavior of elements in the architecture (assuming a successful connection request, without remediation). Each specific deployment of the architecture will have its own unique policy configuration and network topology aspects that will dictate how additional steps may occur.