AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY
Department of Communications and Networking

Markus Pitkäranta

NETWORK ACCESS CONTROL BASED ON ENDPOINT INTEGRITY - INDUSTRY STANDARDS AND COMMERCIAL IMPLEMENTATIONS

Thesis submitted for examination for the degree of Master of Science in Technology

Espoo 1.2.2010

Thesis supervisor:

Prof. Jukka Manner

Thesis instructor:

Thomas G. Jørgensen, Ph.D.

Author: Markus Pitkäranta

Title: Network Access Control Based on Endpoint Integrity
- Industry Standards and Commercial Implementations

Date: 1.2.2010 Language: English Number of pages: 10+64

Department: Department of Communications and Networking

Professorship: Networking Technology Code: S-38

Supervisor: Prof. Jukka Manner

Instructor: Thomas G. Jørgensen, Ph.D.

Network security is an essential part of designing today's corporate networks. Traditionally security threats have been addressed by using network segmentation, firewalls, intrusion detection systems and so forth. However, most of the networks are still vulnerable to attacks coming from inside the internal network. Users in enterprise environments are becoming increasingly mobile when desktop computers are changing to portable computers and handheld devices. From a security perspective this poses new threats. The devices are moved outside the secure corporate network and connected to insecure networks in airports, hotels, cafés, etc. Their security software that defends from malicious users might not be up to date which may expose the device to infection. When the device is connected back to the corporate environment, the whole network might become under threat.

Network Access Control based on Endpoint Integrity is a set of mechanisms to enforce security policies for network devices. The idea is that network access is granted only after certain compliance checks have been passed. Non-compliant endpoints can be denied access or they can be isolated into a dedicated network segment where they can be remediated. Remediation is the process where a non-compliant node is made compliant by applying necessary changes into configurations, installing the latest virus signatures, etc.

Keywords: Authentication, Authorization, Access Control, Trusted Network Connect

AALTO-YLIOPISTON TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Markus Pitkäranta

Työn nimi: Päätelaitteen eheyteen pohjautuva verkon pääsynvalvonta - Teollisuusstandardit ja kaupalliset toteutukset

Päivämäärä: 1.2.2010 Kieli: Englanti Sivumäärä: 10+64

Osasto: Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Jukka Manner

Ohjaaja: TkT Thomas G. Jørgensen

Tietoturva on keskeinen osa nykyaikaista verkkosuunnittelua. Perinteisesti tietoturvaa on pyritty parantamaan käyttämällä mm. verkkojen segmentointia, palomuureja sekä erilaisia IDS/IPS-järjestelmiä. Ongelma nykypäivän organisaatioissa on yhä enemmän ja enemmän liikkuvat käyttäjät. Kannettavat tietokoneet ovat syrjäyttäneet perinteiset pöytätietokoneet, mikä tuo uusia riskejä tietoturvanäkökulmasta sillä laitteet liikkuvat suojatun yritysverkon ulkopuolelle. Käyttävät kytkevät laitteita julkiseen verkkoon lentokentällä, kahviloissa sekä hotellien aulassa. Julkisissa verkoissa koneet altistuvat helpommin hyökkäyksille. Mikäli laitteen tietoturva-asetukset eivät ole ajan tasalla tai esimerkiksi palomuuri on kytketty pois päältä, laite saattaa saada tartunnan. Siinä vaiheessa kun saastunut kone kytketään takaisin yrityksen sisäverkkoon, tartunta saattaa levitä koko verkon laajuisesti.

Päätelaitteen eheyteen pohjautuva verkon pääsynvalvonta on joukko mekanismeja, joiden avulla päätelaitten tietoturva-asetukset voidaan pakottaa määritettyjen tietoturvakäytäntöjen mukaisiksi. Laitteen muodostaessa yhteyden verkkoon sille tehdään tietyt tarkistukset, joiden pohjalta päätetään sallitaanko laitteen pääsy verkkoon. Laitteet, jotka eivät vastaa tietoturvakäytäntöjä, voidaan eristää erilliseen karanteeniverkkoon, jossa laitteiden asetukset voidaan palauttaa käytäntöjen mukaisiksi esimerkiksi asentamalla uusimmat virustunnisteet.

Avainsanat: Authentication, Authorization, Access Control, Trusted Network Connect

Preface

Integrity-based network access control has received a lot of attention during the last few years. This thesis was carried out to give a better understanding of the current status of the industry, and identify the key players in the market. The work was supported by Avanade which is a global IT consulting company specialized in Microsoft technologies.

I would like to thank my istructor Thomas G. Jørgensen for his constructive comments and contribution to this thesis. I will buy you a Tuborg the next time I am visiting Copenhagen.

Otaniemi, 1.2.2010

Markus V.J. Pitkäranta

Contents

Abbreviations

Abbreviations

AAA	Authentication, authorization and accounting
ACE	Access Control Entity
ACL	Access Control List
AD	Active Directory
AIK	Attestation Identity Key
API	Application Programming Interface
AR	Access Requestor
ASD	Asserting Security Domain
CA	Certificate Authority
CNAC	Cisco Network Admission Control
CoA	Change-of-Authorization
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EAP	Extensible Authentication Protocol
EAPOL	EAP over LAN
EC	Enforcement Client
EK	Endorsement Key
ES	Enforcement Server
HRA	Health Registration Authority
IdP	Identity Provider
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IF-FTNC	Federated TNC protocol
IF-IMC	Integrity Measurement Collector Interface
IF-IMV	Integrity Measurement Verifier Interface
IF-M	Vendor-Specific IMC-IMV Messages
IF-MAP	Metadata Access Protocol
IF-PTS	Platform Trust Services Interface
IF-T	Network Authorization Transport Protocol
IF-TNCCS	TNC Client-Server Interface
IF-TNCCS-SOH	IF-TNCCS bindings for Microsoft SoH protocol
IMC	Integrity Measurement Controller
IMV	Integrity Measurement Verifier
IPS	Intrusion Prevention System
IWG	TCG Infrastructure Work Group
MAC	Media Access Control
MAC	Message Authentication Code
MAP	Metadata Access Point
MAPC	MAP Client
MAPC	MAP Client

NAA	Network Access Authority
NAC	Network Access Control
NAC-EI	Network Access Control Based on Endpoint Integrity
NAP	Network Access Protection
NAR	Network Access Requestor
NEA	Network Endpoint Assessment
NEA-WG	Network Endpoint Assessment Working Group
NPS	Network Policy Server
OTP	One-Time Password
PA	Posture Attribute Protocol
PB	Posture Broker Protocol
PCR	Platform Configuration Register
PDP	Policy Decision Point
PEAP	Protected EAP
PEP	Policy Enforcement Point
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
PRA	Provisioning and Remediation Application
PRR	Provisioning and Remediation Resource
PT	Posture Transport Protocol
PTS	Platform Trust Services
RADIUS	Remote Authentication Dial In User Service
RSD	Relying Security Domain
SAML	Security Assertion Markup Language
SHA	System Health Agent
SHV	System Health Validator
SKAE	Subject Key Attestation Evidence
SNAC	Symantec Network Access Control
SOAP	Simple Object Access Protocol
SoH	Statement of Health
SoHR	Statement of Health Response
SP	Service Provider
SPI	Security Posture Information
SSoH	System Statement of Health
SSoHR	System Statement of Health Response
TCG	Trusted Computing Group
TLS	Transport Layer Security
TLV	Type-length-value
TNC	Trusted Network Connect
TNCC	TNC Client
TNCS	TNC Server
TPM	Trusted Platform Module
TS	Terminal Server

TSS	TCG Software Stack
UAC	Unified Access Control
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
VSA	Vendor-Specific Attribute
XML	Extensible Markup Language

1 Introduction

Network Access Control (NAC) is a general terminology for mechanisms that are used in computer networks to restrict access to network resources based on security policies. The traditional methods of controlling access to a network have been authentication and authorization. Authentication is the process of verifying that the client is actually who it claims to be, and authorization means granting or denying access to resources, often based on identity information received from the authentication process.

In modern times computers have gotten increasingly mobile, which has led to new kinds of security threats from a networking perspective. Traditionally the biggest challenge of network operators has been blocking attacks from malicious clients by using such mechanisms as NAC described above and other network security mechanisms, such as network segmentation, firewalls and Intrusion Detection and Prevention Systems (IDS/IPS). The problem with these mechanisms is that they usually assume that the attack comes from a malicious client. The problem with mobile computers and users is that the computers are connected into public networks in airports, cafés, etc. The clients usually have some kind of client security software installed to protect them from security threats. However there will be times when the client security software is turned off due to a user being frustrated by it consuming too many resources, or the operating system or software program on the client is out of date because it has not received the latest security updates. Any of these scenarios may lead into a situation where the client is exposed to security threats. If the client gets infected and then connects to the internal network, the infection may propagate to other clients within the network. In this situation the attack is coming from what seems to be a legitimate client. Internal networks are often considered more secure than public networks and the clients may be configured with less strict security settings which makes them vulnerable. This can even strengthen the possible impact of the infection on the network.

One possible resolution for the problems described above would be to make the access control decision based on the endpoint's integrity or security posture. The integrity is a collection of security measurements that define the security state of the client, and these integrity measurements can be compared to a set of security policies predefined to decide whether the client is compliant and granted access to the network. Possible integrity measurements could include the state of a client firewall or antivirus software and whether the client's operating system is up to date.

At the time of writing, there are several commercial products for integrity-based network access control. Such vendors as Cisco, Juniper, HP, Symantec and Microsoft have brought their solutions into market but currently the architecture lacks standardization which makes the implementations of different vendors unable to inter-operate. Trusted Computing Group (TCG) is a non-profit organization formed to promote vendor-neutral industry standards for secure computing [2]. TCG has developed an architecture for integrity-based network access control called Trusted Network Connect (TNC). TNC is an attempt at interoperability between the implementations of different vendors, but unfortunately currently most of the implementations are not compatible with the TNC.

The Internet Engineering Task Force (IETF) has also a working group defining standards for integrity-based network access control. IETF Network Endpoint Assessment (NEA) is a working group formed to agree standards for this area. IETF NEA has drafted a requirements document for reference architecture [4] and is going to solicit candidate protocol specifications and evaluate them as candidate specifications. Currently there are drafts for two of the protocols in the NEA reference architecture and they are derived from the TNC architecture.

The goal of this thesis is two-fold: The primary goal is to analyze how a current commercial implementation conforms to one of the proposed vendor-neutral standards. The secondary goal is to evaluate the future of Network Access Control Based on Endpoint Integrity (NAC-EI). Thus the starting point for this study is to describe an industry standard architecture for NAC-EI and then compare it to a commercial implementation. The industry standard chosen for this study is the TNC architecture, because it is currently the most comprehensive standard available. Microsoft Network Access Protection (NAP) will be compared to TNC because NAP and TNC are claimed to interoperate [1]. Based on this analysis, an evaluation of the discussion of NAC-EI implementations is provided.

The emphasis in this thesis is on the architectural design of the NAC-EI implementations. Thus the major achievement is to provide an overview of the two architectures described. TNC and NAP share a lot of similarities as evidenced by the comparison of these frameworks. This is not a coincidence as Microsoft is one of the contributors of the TNC Work Group. When analyzing the current market situation we saw that the main competing frameworks are the TNC and Cisco Network Admission Control (CNAC). The conclusion was that out of these two the TNC is becoming the leading framework for NAC-EI. The main reason for this is the fact that TNC is becoming an IETF standard.

This thesis is divided into three parts. Chapter 2 describes the TNC architecture in detail. In chapter 3 we evaluate the Microsoft NAP architecture and compare it to TNC. Chapter 4 provides a brief analysis of the future of NAC-EI standards and implementations.

2 Industry Standards for NAC-EI

This chapter presents an overview of current industry standards for Network Access Control based on Endpoint Integrity (NAC-EI). The chapter focuses on Trusted Network Connect (TNC) by describing the building blocks of the architecture.

2.1 Background

This section is an overview of the current industry standards and work groups defining architectures for NAC-EI.

2.1.1 Trusted Network Connect (TNC)

Trusted Computing Group is a non-profit organization founded in 2003 to develop and promote vendor-neutral open standards for computer security. The TCG has members and contributors from industry leading companies, such as Hewlett-Packard, IBM, Juniper and Microsoft. The standards involve hardware building blocks and software interface specifications across multiple hardware platforms and operating systems. The most widely implemented TCG standard is the Trusted Platform Module (TPM) which is a hardware module used for storing cryptographic keys. Most of today's laptops, especially the ones designed for business users, have a TPM chip installed. The TPM has been also accepted as an ISO/IEC standard [27]. [2]

Trusted Network Connect (TNC) is an architecture developed by the TCG that enables access control based on endpoint integrity in addition to traditional authentication and authorization. The goal of TNC is to enable network operators to define policies regarding the security state of endpoints. The endpoint's compliance with these security policies can then be used to determine whether to grant or deny access to a network resource. The TNC architecture will be described more granularly later in this study. [3]

2.1.2 Network Endpoint Assessment (NEA)

The Internet Engineering Task Force (IETF) is also working on standardizing protocols for NAC-EI. The architecture is referred as Network Endpoint Assessment (NEA) and the work group defining the architecture is Network Endpoint Assessment Working Group (NEA WG). NEA uses the term ”posture” to refer to the security state of an endpoint. The posture of an endpoint is derived from such information as hardware and software configuration including whether a software component installed to protect the endpoint (e.g., an anti-virus program or client firewall) is up-to-date. The goal of NEA is to provide a standard way of assessing the posture of an endpoint for the purpose of monitoring compliance to the posture policy of the organization. [4]

The NEA supports two deployment scenarios: advisory mode and mandatory mode. In advisory mode the endpoint is granted unrestricted access to the network regardless of the

result of the posture assessment. This deployment scenario can be used for just monitoring the posture of the endpoints. In mandatory mode the endpoint gains only restricted access to support remediation purposes, i.e. to bring the endpoint compliant with the posture policy. Defining mechanisms for providing restricted access is out of the scope of the NEA WG. [4]

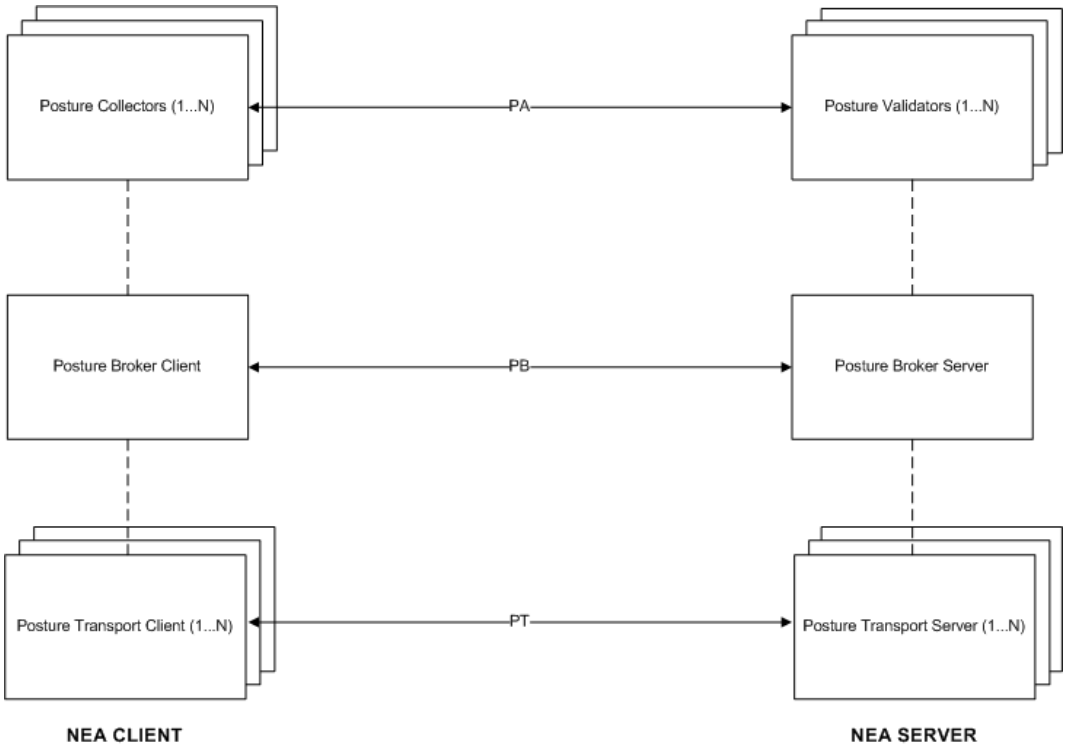

Figure 1: NEA Reference Model [4]

Figure 1 describes the reference model for Network Endpoint Assessment. The NEA Client is comprised of three kinds of components: Posture Collectors, Posture Broker Client and Posture Transport Clients. These components communicate with their counterparts on the NEA Server. The Posture Collectors exchange posture information with the Posture Validators using the Posture Attribute Protocol (PA). The Posture Broker Client and Posture Broker Server multiplex and de-multiplex these messages and communicate with each other using the Posture Broker Protocol (PB). The Posture Transport Client is responsible for establishing a reliable communication channel with the NEA Server. There may be multiple Posture Transport Clients within a particular Posture Broker Client to support multiple protocols. Each Posture Transport Client communicates with a Posture Transport Server using a Posture Transport Protocol (PT). Defining the PT is out of the scope of the NEA WG as there are existing protocols that can be used for the PT. The dashed lines in Figure 1 represent the Application Programming Interfaces (APIs) and/or protocols between the components within the NEA Client or NEA Server. Standardizing these interfaces is also out of scope of the NEA WG.

When standardizing the Posture Attribute Protocol and Posture Broker Protocol the NEA WG has recognized that there already exist several non-standard protocols that may fill

the requirements for the PA ad PB. Currently the NEA WG has accepted the IF-M and IF-TNCCS from the TNC architecture as candidate protocols [5] [6].

2.2 TNC Architecture

The Trusted Network Connect (TNC) architecture is based on the TCG Infrastructure Work Group (IWG) Reference Architecture for Interoperability which is an architecture aiming at improving interoperability between systems containing TCG technology [7]. One of the most important concepts in the architecture is Trusted Platform, which refers to hardware-protected platform authentication. In practice this is equivalent of using a Trusted Platform Module [27], which is a computer chip that can store securely artifacts that can be used for platform authenticaion.

The IWG Platform-Authentication model defines three entitites: Requestor, Verifier and Relying Party as depicted in Figure 2. Requestor is the platform seeking to be authenticatied by the verifier. The Requestor is performing a transaction with the Relying Party which relies on the Verifier to evaluate the statements issued by the Requestor. The Verifier performs the evaluation of the Requestor's statements based on some set of criteria or rules. One example of this kind of a network access control architecture is the IEEE 802.1X. In this example the requestor would be an 802.1X Supplicant, the relying party would be an 802.1X Authenticator (e.g., switch) and the verifier would be a 802.1X Authentication Server. [8] [3]

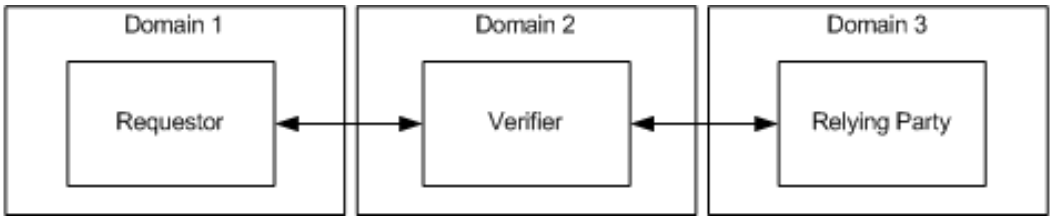

Figure 2: The IWG Platform-Authentication model

In the TNC architecture, the three main components are Access Requestor (AR), Policy Decision Point (PDP) and Policy Enforcement Point (PEP) as shown in Figure 3. The AR is equivalent to Requestor in the IWG Architecure, the PDP maps to Verifier and the PEP acts as the Relying Party. The naming convention is quite describing as the Access Requestor is the client requesting for access to the network. The Policy Decision Point is the party that makes the actual access decision, and Policy Enforcement Point is the component that enforces the decision made by the PDP. [3]

Figure 4 shows a more detailed view of the TNC Architecture. The entities are depicted by the five columns in the figure. The different entities are the Access Requestor (AR), the Policy Enforcement Point (PEP), the Policy Decision Point (PDP), the Metadata Access Point (MAP) and Flow Controllers and Sensors. The entities can be divided into three layers which distinguish different components within each entity. The interfaces between the components are depicted with the dashed lines and arrows.

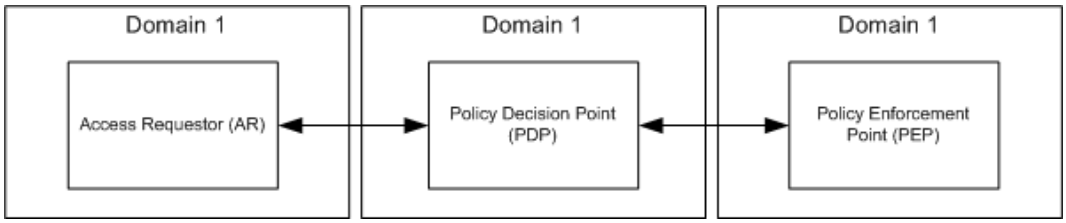

Figure 3: A simplified view of the TNC Architecture

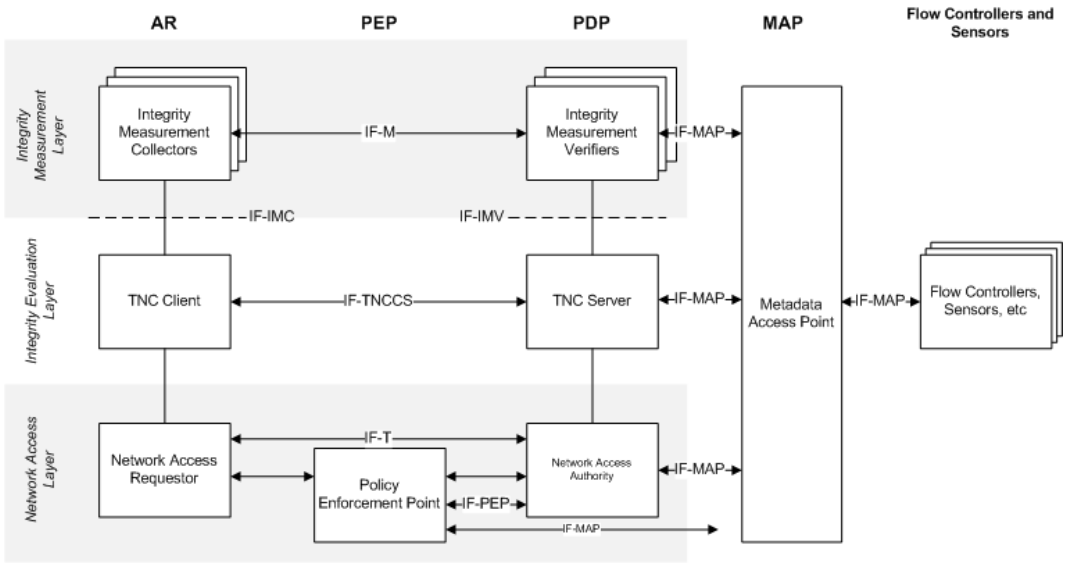

Figure 4: The TNC Architecture

2.3 TNC Layers

The TNC architecture divides into three layers: network access layer, integrity evaluation layer and integrity measurement layer. The network access layer is responsible for network connectivity. This can involve several network technologies such as Virtual Private Networking (VPN) and 802.1X. The three components within the network access layer are Network Access Requestor (NAR), Policy Enforcement Point (PEP) and Network Access Authority (NAA). [3]

The integrity evaluation layer contains components that are responsible for evaluating the overall integrity of the Access Requestor. The components on the integrity evaluation layer use the information received from the integrity measurement layer as input. The two components within the integrity evaluation layer are the TNC Client and the TNC Server. The integrity measurement layer contains the components that collect and verify integrity information, i.e., Integrity Measurement Collectors (IMCs) and Integrity Measurement Verifiers (IMVs). [3]

2.4 TNC Entities and Components

The entities and components in the TNC Architecture are logical, not physical. A component or an entity may be a software program, a physical device or a replicated set of machines to serve redundancy. The entities can be divided into required entities and optional entities. The required entities are the Access Requestor (AR) and the Policy Decision Point (PDP). The optional entities are the Policy Enforcement Point (PEP), Metadata Access Point (MAP), and Flow Controllers and Sensors. [3]

Access Requestor (AR) is the entity seeking access to a protected network in order to conduct activities in the network. The AR consists of three components: Network Access Requestor (NAR), TNC Client (TNCC) and Integrity Measurement Collector (IMC). NAR is the component responsible for establishing network access with the PEP and PDP. There may be several NARs within a single AR to support various connection mechanisms. Examples of NARs include VPN client software and IEEE 802.1X client. [3]

The TNCC is a software component that aggregates the integrity measurement information received from the IMCs and communicates the integrity state of the client to the TNC Server. There may be multiple IMCs within an AR and each IMC measures some aspect of the endpoint posture. For example, one IMC can check the status of a client anti-virus software while another IMC may check that the operating system has the latest security updates installed. Because this kind of information often depend on the implementation, vendors can implement their own IMCs to provide that kind of integrity measurements. [3]

Policy Decision Point (PDP) makes the decision regarding the Access Requestor's network access request. The decision is made using the integrity information provided by the AR and a predefined set of security policies. The PDP consists of three components: Network Access Authority (NAA), TNC Server (TNCS) and Integrity Measurement Verifier (IMV). The NAA is the component that decides whether the AR should be granted or denied access to the network. The NAA makes the decision based on the information it gets from the TNCS. [3]

The TNCS is the component that manages the communication between the IMCs and IMVs. The TNCS aggregates the Action Recommendations from the IMVs and combines them into an overall Action Recommendation and sends it to the NAA. The IMV is a software component that verifies some aspect of the integrity of the client. [3]

Policy Enforcement Point (PEP) is the name of the entity and component that executes the network access decision made by the PDP. This decision might be to allow unrestricted access to the network for the AR, to allow only restricted access, or to deny access. Examples of Policy Enforcement Points include network devices such as switches or wireless access points.[3]

Metadata Access Point (MAP) is the name of the entity and component that allows other components to publish, subscribe and query information about the integrity state of the ARs. This allows components such as flow controllers and sensors to use the integrity information, as well as other information about the endpoints. [3]

Flow controllers are components that control the flows in the network based on information obtained from the MAP. Examples of flow controllers include firewalls and rate limiters that can block or limit the network traffic originated from or destined for a particular endpoint or user. Sensors are components that publish information about network activities to the Metadata Access Point. Examples of such activities include an endpoint accessing certain services in the network, authentication activities and broadcast messages for services (e.g., DHCP). [3]

2.5 TNC Interfaces

This section describes the interfaces and protocols defined in the TNC architecture.

2.5.1 Integrity Measurement Collector Interface (IF-IMC) and Integrity Measurement Verifier Interface (IF-IMV)

Integrity Measurement Collector Interface (IF-IMC) is the interface between the Integrity Measurement Collectors and the TNC Client as depicted in Figure 5. Using the IF-IMC, the TNC Client can collect integrity measurement information from the IMCs. The IF-IMC is closely related to Integrity Measurement Verifier Interface (IF-IMV) which is the interface between the Integrity Measurement Verifiers and the TNC Server.[9] [10]

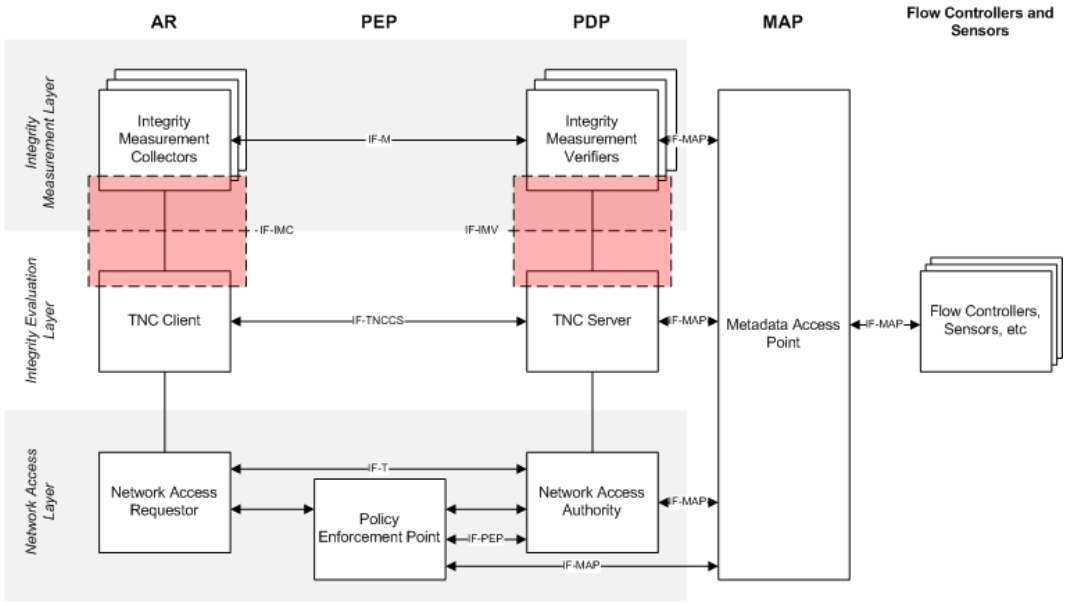

Figure 5: The IF-IMC and IF-IMV

The supported use cases for IF-IMC[9]:

- A TNCC and one or more IMCs that support the same platform binding have been installed on an endpoint. The TNCC finds and loads the IMCs. Then it runs one

or more Integrity Check Handshakes. The IMCs and TNCC may use any of the features of IF-IMC.

- A TNCC that supports the Java Platform Binding has restricted privileges/permissions (as when loaded into a sandbox in a web browser). It loads IMCs that support the Java Platform Binding and runs one or more Integrity Check Handshakes. The IMCs may actually have greater privileges than the TNCC (or they may not be sandboxed).
- A TNCC that supports the Java Platform Binding runs with generous privileges but chooses to run IMCs with restricted privileges for security reasons. It loads IMCs and runs one or more Integrity Check Handshakes.
- A TNCC is running on an endpoint. When an IMC is installed or uninstalled, the TNCC notices this and loads or unloads the IMC.

The supported use cases for IF-IMV are [10]:

- An IMV and TNCS that support the same platform binding are installed on an endpoint. The TNCS finds and loads the IMV. Then it runs one or more Integrity Check Handshakes. The IMV and TNCS may use any of the features of IF-IMV.
- A TNCS has restricted privileges. It loads IMVs and runs one or more Integrity Check Handshakes.
- A TNCS that supports the Java Platform Binding runs with generous privileges but chooses to run IMVs with restricted privileges for security reasons. It loads IMVs and runs one or more Integrity Check Handshakes.
- An IMV and a TNCS both support the reason string extensions to IF-IMV. An IMV provides a reason string to a TNCS, giving the reason for its IMV Action Recommendation. The TNCS logs this reason and/or passes it on to the TNCC through an extension to IF-TNCCS. At the TNCC, the reason information may be displayed to the user (perhaps in a detailed view). The reason string may be in the endpoint user's preferred language or (if that language is not available) in another language.
- A TNCS is running. When an IMV is installed or uninstalled, the TNCC notices this and loads or unloads the IMV.

IF-IMC and IF-IMV contain the following features: Integrity Check Handshake, Connection Management, Remediation and Handshake Retry, Message Delivery and Batches. Integrity Check Handshake is the context that IMCs and IMVs use to exchange messages with each other. This is a very important functionality as this enables IMVs to become aware of the security state of the IMCs. This information exchange is the foundation for the access control decision made by the PDP. In the Integrity Check Handshake the IMCs send a batch of messages to the IMVs, and the IMVs may respond with a batch of messages. The response may contain queries for more information, remediation instructions,

etc. In the end of this dialog the IMVs make their decision on their IMV Action Recommendations. Several Integrity Check Handshakes may occur between a TNCC and a TNCS. The first handshake may end with the client receiving remediation instructions to become compliant. After the remediation another handshake is performed to prove the clients compliance, and subsequent handshakes might occur e.g., in the case of a policy change. For a given TNCC-TNCS pair one handshake must always end before another one begins. [9] [10]

The TNCC maintains the connection state by using a connection ID, which is an identifier chosen when the connection is established. The connection ID is completely local for the TNCC, so it is not shared with the TNCS. When a new network connection is established the TNCC notifies the IMCs of the new connection and informs them if the state of the connection is changed. When the connection ends the TNCC informs the IMCs that the connection is terminated and that the connection ID will be deleted. However, the TNCC should use the same connection ID while communicating with the same TNCS. This allows the TNCC to request a handshake retry for an earlier connection, e.g., after it has completed remediation. The remediation might require restarting the operating system or power cycling, so the connection ID should be retained even during these operations in order to be able to request the handshake retry. [9] [10]

There are a few scenarios when it is desirable to retry the Integrity Check Handshake without establishing a new network connection with a new connection ID. If an endpoint has been isolated and and then remediated an IMC can suggest the TNCC to retry the Integrity Check Handshake with the existing connection ID without establishing a new connection. The TNCS can also initiate a handshake retry as a result of a security policy change or as a periodic activity. And finally, an IMC or IMV can request a handshake retry in response to a condition detected by the IMC or IMV. [9]

The network access decision in the TNC architecture is based on information exchange between IMCs and IMVs. That is why the message delivery between IMCs and IMVs a critical function for the architecture. IF-IMC includes functions for sending and receiving those messages. The message itself consists of a message body, a message type, and a recipient type. The message body is sequence of bytes. The message type is a number that uniquely defines the structure and semantics of the message body. The recipient type is simply a flag that indicates whether the recipient is an IMC or an IMV. The TNCC and TNCS should not parse the message body as it is done by the IMCs and IMVs. The IMCs and IMVs announce to the TNCC and TNCS which message types they want to receive. The TNCC and TNCS use the message type and recipient type to route the message to correct recipient or recipients. [9] [10]

The messages between IMCs and IMVs are always carried over in batches. This is due to the fact that these messages are often carried over protocols like Extensible Authentication Protocol (EAP) that require endpoints to take turns in sending messages i.e., only one endpoint can send at the time. The first batch of messages is always sent by the IMC. The IMV can respond to these messages and the IMCs can respond to the messages sent by the IMV. This way the dialog goes on until the handshake is over. [9] [10]

2.5.2 TNC Client-Server Interface (IF-TNCCS)

The IF-TNCCS defines a protocol for exchanging messages between the TNC Client and TNC Server as described in Figure 6. The types of messages include [11]:

- Messages from IMCs to IMVs (e.g., integrity measurements)
- Messages from IMVs to IMCs (e.g., remediation instructions or requests for additional measurements)
- Messages from TNCCs to TNCSs (e.g., control messages)
- Messages from TNCSs to TNCCs (e.g., TNCCS-Recommendation message)

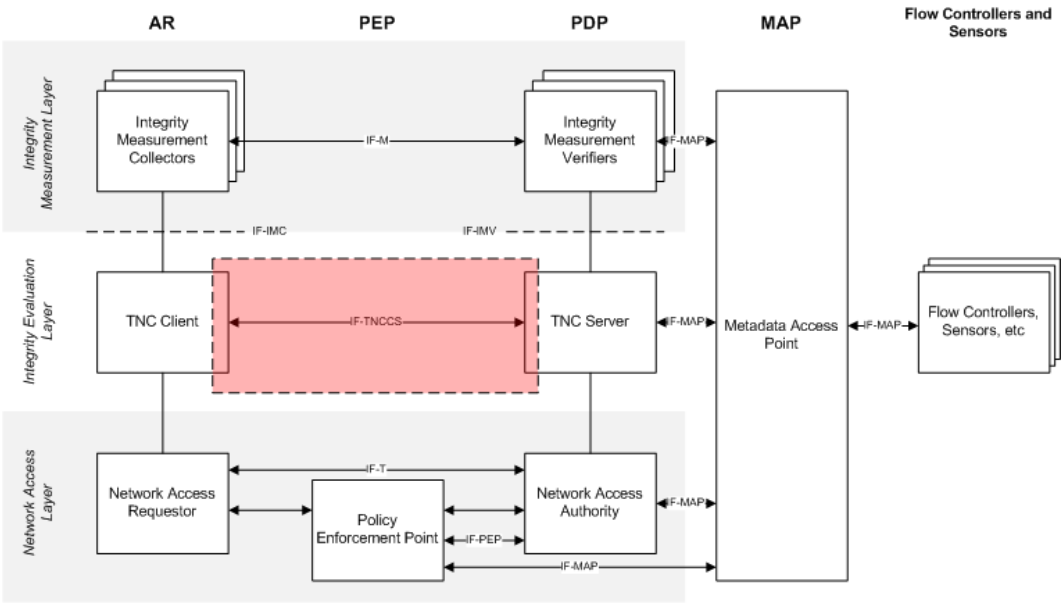

Figure 6: The IF-TNCCS

Supported use cases for IF-TNCCS are:

- The messages sent from IMCs to IMVs during the the Integrity Check Handshake are carried through IF-TNCCS.
- During the Integrity Check Handshake the IMVs may optionally respond to the IMCs (e.g., remediation instructions). These messages are carried through IF-TNCCS.
- The TNCC may send a batch of messages to the TNCS by using IF-TNCCS. These messages can be either standardize or vendor-specific.
- The TNCS may send a batch of messages to the TNCC by using IF-TNCCS. These messages can be either standardize or vendor-specific.

- IMVs may provide a reason string describing the reason for their IMV Action Recommendations. The TNCS may pass these reason strings to the TNCC by using an extension to IF-TNCCS.
- The TNCS may inform the TNCC of its IP address and port number.

One of the primary functions of IF-TNCCS is to facilitate message exchanges between the IMCs and IMVs so that the IMVs can TNC Server can assess the security state of the TNC Client. The communication between the IMCs and IMVs is always performed within the context of an Integrity Check Handshake, which was described earlier in context of the IF-IMC and IF-IMV. [11]

The messages exchanged between the TNC Client and TNC Server contain a message type, which is a four octet number uniquely identifying the format and semantics of the message. To ensure uniqueness the message type is formed from a vendor ID and a message subtype. The first three octets reflect the vendor ID, and the last octet describes a vendor-chosen message subtype. SMI Private Enterprise Numbers are used to provide the vendor IDs [14]. [11]

The IF-TNCCS contains a set of standardized TNCC-TNCS messages. These messages are listed in table 1. All the standardized messages have the vendor ID of zero.

Table 1: TCG Standardized TNCC-TNCS Messages [11]

Message Type	**Value**
TNCCS-Recommendation	0x00000001
TNCCS-Error	0x00000002
TNCCS-PreferredLanguage	0x00000003
TNCCS-ReasonStrings	0x00000004
TNCCS-TNCSContactInfo	0x00000005

The TNC Server sends a TNCCS-Recommendation to inform the TNC Client that the Integrity Check Handshake has completed and that the TNCS is ready to provide the TNCS Action Recommendation (allowed, isolated or none) for the NAA. The TNCCS Action Recommendation is included in the message. As the name implies, it is only a recommendation, and the NAA may ignore the decision made by the TNC Server and apply a different level of access. [11]

A TNCCS-Error message is sent when there has been a problem with the previous batch of messages. If the error message is sent from a TNC Client to the TNC Server then it must be in a batch that doesn't contain any IMC-IMV messages. If it's sent from the TNC Server to the TNC Client it must be placed into the same batch with the TNCS Action Recommendation message. [11]

The TNC Client can send a TNCCS-PreferredLanguage to indicate its preferred language to be used. This information is particularly useful when the IMV sends its reason strings to the TNC Server subsequently to the TNC Client. The reason strings are included in a TNCCS-ReasonStrings message, which is sent from the TNC Server to the

TNC Client to explain the reason of the TNCS Action Recommendation. The TNCCS-PreferredLanguage can contain multiple reason strings each of which should be tagged with an RFC 3066 language tag [12] to indicate the language used. [11]

At the end of a successful assessment the TNC Server can use a TNCCS-TNCSContactInfo message to inform the TNC Client of its IP address and port number. The TNC Client can use this information to contact the TNC Server over L3 after it has been granted access to the network. The use of TNCCS-TNCSContactInfo supports and enables the use if IF-T over TLS [26] where the TNC Client is provided the IP address and port number of the TNC Server during a layer 2 IF-T assessment and then uses that address and port number to connect to the TNC Server. [11]

The messages between IMCs and IMVs are routed using two fields in the message: message type and recipient type. Each IMC and IMV indicates which message types it wants to receive. The TNCC and TNCS delivers the messages to those IMCs and IMVs that have the recipient type and corresponding to the recipient type of the message and that have announced to be willing to receive messages with the message type of the message. [11]

The IMC-IMV and TNCC-TNCS messages are always sent in batches. The Integrity Check Handshake starts with a batch of messages sent by the TNC Client to the TNC Server. The IMC-IMV messages are delivered to the IMVs that have registered for the corresponding message types. The TNCC-TNCS messages are parsed by the TNCS. If the IMVs or the TNCS want to response to the messages received, the TNCS gathers these messages into a batch and sends it to the TNCC. Again, the TNCC delivers the IMC-IMV messages to the corresponding IMCs and pops any TNCC-TNCS messages. This dialog goes on until the handshake is completed. If none of the IMCs want to send a message in a particular batch batch the TNCC will complete the handshake by sending a batch containing no IMC-IMV messages. This indicates to the TNCS that the TNCC has no more measurements to provide and that it wants to complete the handshake. Similarly, if none of the IMVs have messages to send in a batch, this indicates to the TNCS that the IMVs are ready to give their IMV Action Recommendations. The TNCS gathers these recommendatios into a TNCS Action Recommendations which is sent to the TNCC in a batch containing no IMC-IMV messages. [11]

If the TNC Server receives a malformed batch, or has an internal error, it should discard the batch and generate a TNCCS-Recommendation message containing a TNCCS-Error message and send it to the TNC Client. Similarly, if a TNC Client receives a malformed batch, or has an internal error, it should send a TNCCS-Error message to the TNC Server. When the TNC Server receives the TNCCS-Error message, it should generate a TNCCS-Recommendation message. [11]

IF-TNCCS is based on Extensible Markup Language (XML) and the format of the IF-TNCCS messages is defined in the IF-TNCCS schema [11]. In addition the TNC-WG has defined IF-TNCCS bindings for Statement of Health (SOH) protocol used by Microsoft Network Access Protection (NAP). The goal is to enable interoperability between NAP and systems based on the TNC architecture. [13]

2.5.3 Vendor-Specific IMC-IMV Messages (IF-M)

IF-M refers to the vendor-specific information exchange between IMCs and IMVs as depicted in Figure 7. IF-M is an application level protocol that carries Integrity Check Handshake messages between IMCs and IMVs. The IF-M allows IMCs to send measurement information about the local platform to IMVs for evaluation. The IMVs use IF-M for requesting more measurement information or sending their IMV Action Recommendation to the client side. The recommendation may also contain instructions for remediation. The TCG will standardize certain widely used IF-M messages, but most of the messages will be specific to each IMC-IMV pair, i.e., vendor-specific. [22]

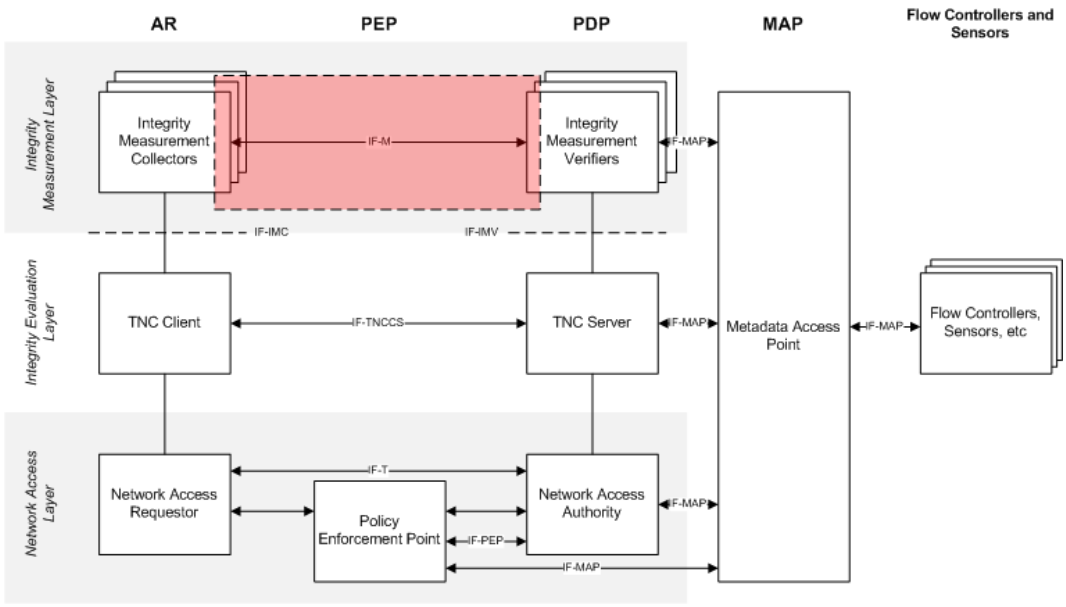

Figure 7: The IF-M

Table 2: IF-M Component Types [22]

Component Name	**Description**
Reserved	Reserved for use in specification examples, experimentation and testing.
Operating System	Operating system running on the endpoint
Anti-Virus	Host-based anti-virus software
Anti-Spyware	Host-based anti-spyware software
Anti-Malware	Host-based anti-malware software
Firewall	Host-based firewall
Intrusion Detection / Prevention (IDS/IPS)	Host-based intrusion detection / prevention software
Virtual Private Networking (VPN)	Host-based VPN software

IF-M messages are transported using IF-TNCCS protocol which provides a reliable end-

to-end delivery to subscribed IMCs and IMVs. IF-TNCCS allows an IMC or IMV to subscribe for a particular type of message indicating that the IMC or IMV is willing to receive messages of that particular type. The message type indicates the software component that is associated with the information contained in the IF-M message and it is used by the TNCC and TNCS to route the messages to the IMCs and IMVs. The message type is comprised of a Vendor ID and a message subtype. Table 2 lists the message subtypes (i.e., component types) currently standardized by the TCG. [22]

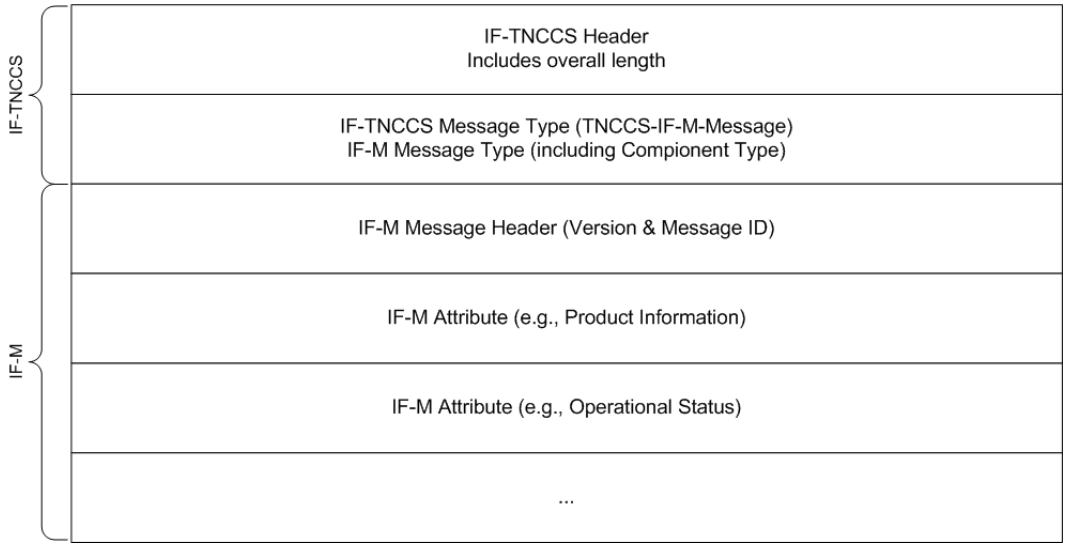

Figure 8: An IF-TNCCS batch containing an IF-M message [22]

Figure 8 depicts an IF-TNCCS batch containing one or more IF-M messages. Each IF-M message consists of a message header followed by zero or more assessment attributes. The IF-M message header and the header for each of the attributes use a fixed type-length-value (TLV) format.

2.5.4 Network Authorization Transport Protocol (IF-T)

Network Authorization Transport Protocol (IF-T) pertains to the message exchange between the Network Access Requestor and Network Access Authority. The TNC is not standardizing IF-T as its own protocol, but provides bindings how to carry these messages over existing lower layer protocols such as Extensible Authentication Protocol (EAP) [15] or Transport Layer Security (TLS) [18]. [3]

Extensible Authentication Protocol (EAP) is an authentication framework that provides an infrastructure for network access clients and authentication servers to host plug-in modules for existing and future authentication methods. EAP was originally developed as an extension to Point-to-Point Protocol (PPP) [17] to support multiple authentication mechanisms without having to pre-negotiate the mechanism during the link establishment phase [15]. EAP is widely used in wireless networks but its support is not limited to wireless networks. The IEEE 802.1X [8] authentication mechanism is based on EAP.

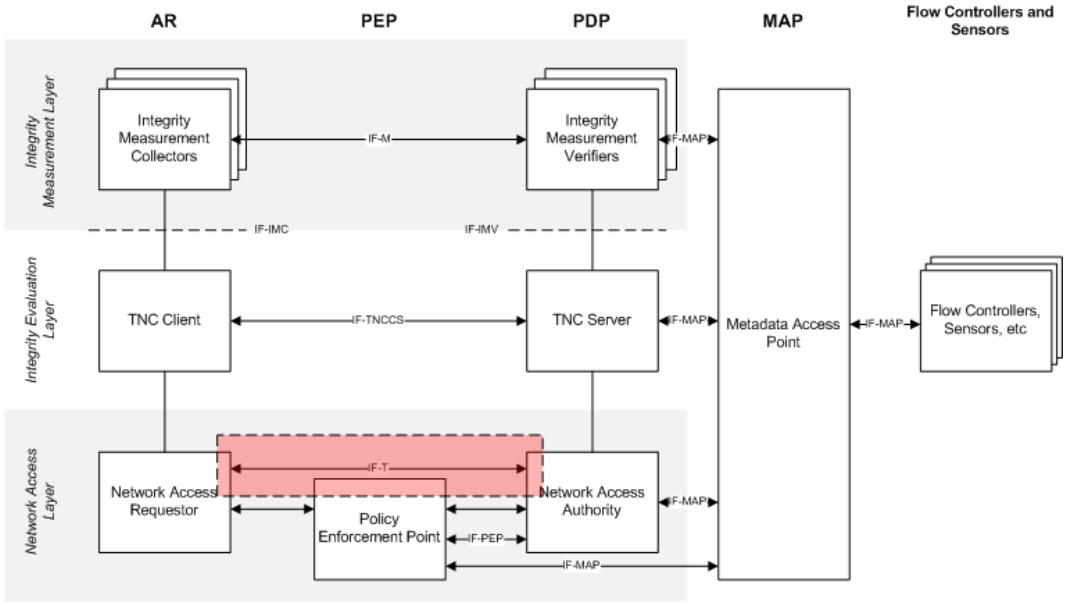

Figure 9: The IF-T

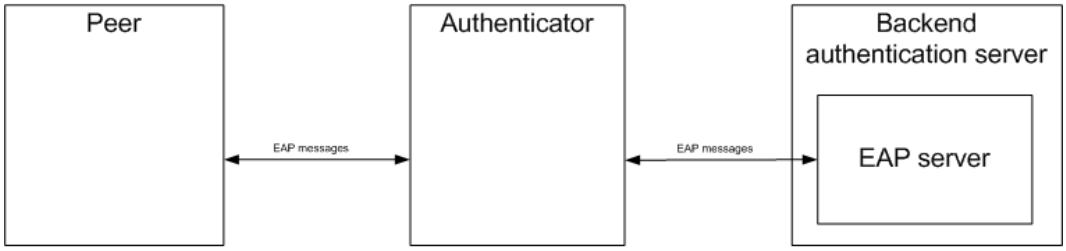

Figure 10: Extensible Authentication Protocol

The three entities participating in EAP authentication are the authenticator, the peer and the authentication server which is often also referred as EAP server. The relation between these entities is depicted in Figure 10. Peer is entity seeking access to a network. In IEEE 802.1X the peer is called the supplicant. Authenticator is the entity requiring the peer to authenticate before gaining access to the network. The authentication server authenticates the peer and provides authentication information to the authenticator. In the case where no backend authentication server is used, the authentication server is part of the authenticator. Buf often the authenticator will send authentication decisions to a backend authentication server in which case the authenticator is operating in pass-through mode. One protocol used to relay EAP messages between the authenticator and authentication server is RADIUS [16]. Figure 11 depicts IEEE 802.1X authentication using RADIUS authentication server. [15] [8]

The EAP has two message types: Requests and Responses. The message exchange be-

gins when the Authenticator sends a Request to the Peer. The Request packet contains a Type field which indicates the type of data being carried. The Type field may indicate authentication mechanism to be used, or it may carry some other information (e.g., declination of a proposed authentication mechanism). The EAP specification defines three authentication types: MD5-Challenge [19], One-Time Password (OTP) [20] and Generic Token Card (GTC). EAP is not limited to theses authentication methods but additional authentication types may be defined. IETF has made a specification of using TLS within EAP [21], and vendors may also define their own authentication types, or plug-in modules to be used within EAP. Multiple authentication methods during an EAP conversation are not supported. But this can be achieved by utilizing a single EAP method and running other methods within it. This is referred as EAP tunneling. [15]

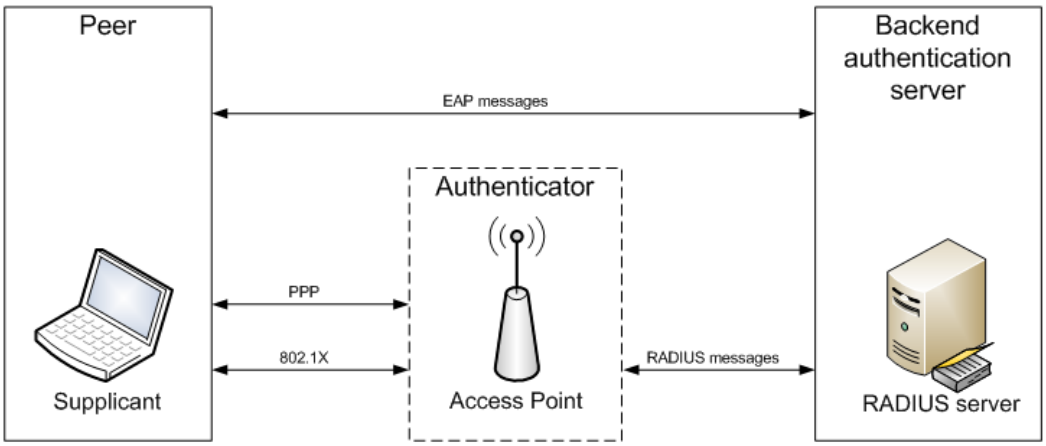

Figure 11: IEEE 802.1X

The TCG has defined protocol bindings for using IF-T over tunneled EAP methods. There are a number of tunneled EAP methods that have been implemented by different vendors. A tunneled EAP method consists of two phases: during the first phase a TLS tunnel is created over EAP, and during the second phase other information is carried over the protected TLS tunnel. The method providing the tunnels is referred as outer method, and the method that is used over the tunnel is called inner method. The function of these tunneled EAP methods is to provide a secure channel for exchanging other authentication and authorization information. By using tunneled EAP methods otherwise insecure EAP methods can be used securely as inner methods, as long as the outer method meets the security requirements. [23]

Figure 12 describes the protocol layers that combine to provide IF-T using tunneled EAP methods. The AAA Server refers to Authentication, Authorization and Accounting Server, which is a general term for a server that controls access to computer resources. The highlighted area indicates the layers that have components that are part of the IF-T protocol binding for tunneled EAP methods. EAP-TNC is a simple EAP method that is used as inner method in the dialog. EAP-TNC encapsulates IF-TNCCS messages they

can be carried over tunneled EAP methods. The EAP-TNC can be carried over any EAP tunneled method as long as both the peer and the authenticator use the same EAP method and implement the conform to the same EAP-TNC standard. This allows a peer from one tunneled method vendor to communicate with an authenticator from a diffenrent vedor. [23]

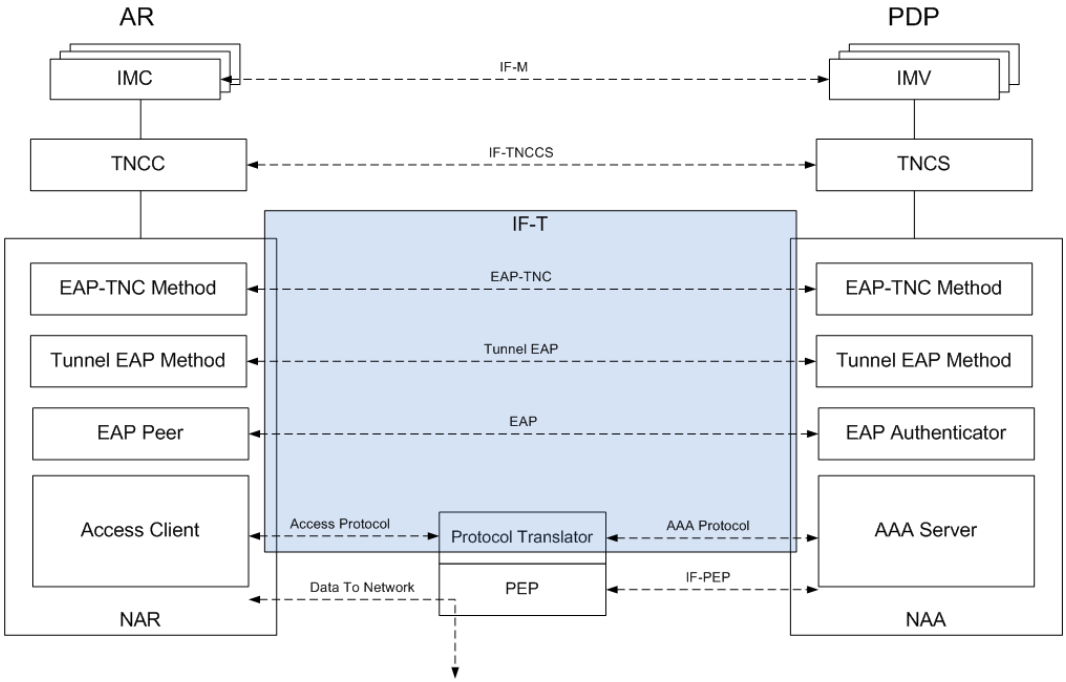

Figure 12: EAP-TNC and EAP Protocol Layers [23]

The Access Client and Protocol Translator initiate the access control dialog by using an access protocol. The Protocol Translator can be a wireless access point, a switch, etc. Examples of access protocols include IEEE 802.1X and IKEv2 [24]. The Protocol Translator and AAA Server exchange messages using an AAA protocol. Examples of AAA protocols include RADIUS and Diameter [25]. There is no limitation for using certain access or AAA protocols, as long as they support EAP authentication. Figure 13 shows how the messages are encapsulated using different protocols. The Protocol Translator removes the Access Protocol headers (e.g., 802.1X or IKEv2) and forwards the EAP message to the NAA using an AAA protocol (e.g., RADIUS). [23]

The TCG also has also defined protocol bindings for using IF-T over TLS. Earlier was described how EAP can be used to carry the assessment information prior to actually connecting to the network, i.e., getting an IP address. In this case the assessment needs to be carried within the protocol that is used during the joining process. But if the client already has an IP address then it can utilize higher layer protocols such as TLS to carry the assessment information. [26]

TLS is a protocol that allows applications to communicate across a network securely. TLS provides authentication, confidentiality and data integrity and it is widely adopted in many services including web servers and e-mail. The TLS protocol is composed of two layers.

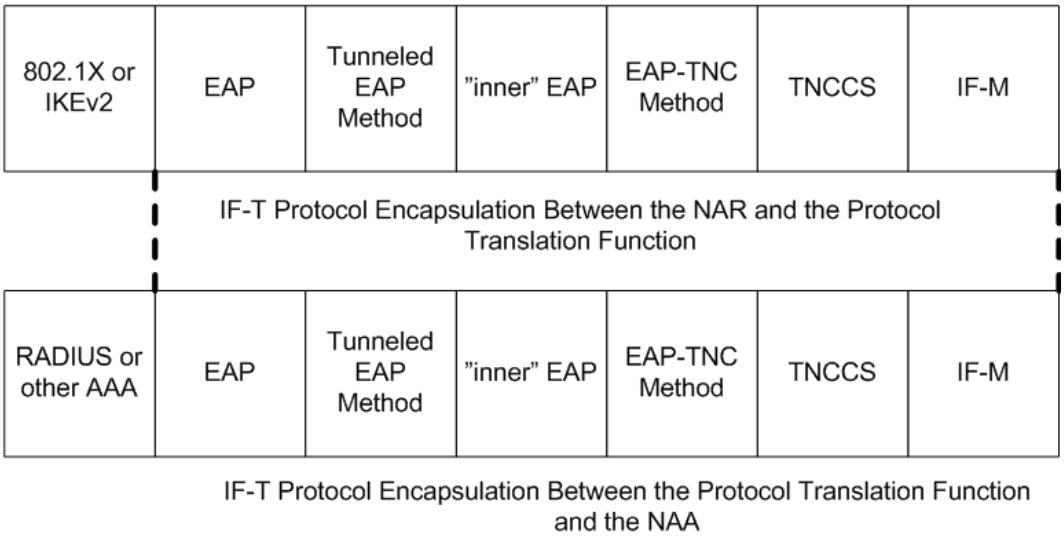

Figure 13: IF-T protocol encapsulation [23]

TLS Record Protocol is the lower level component that operates on top of TCP or some other reliable transport protocol. The TLS Record Protocol provides data privacy and integrity. Data privacy is provided using symmetric cryptography (e.g., AES or RC4). The symmetric encryption keys are negotiated using another protocol (e.g., TLS Handshake Protocol). Data integrity is provided using Message Authentication Codes (MAC). The TLS Record Protocol is used for encapsulating higher-level protocols. The current TLS specification defines four content types or protocols that can be used on top of the Record Protocol: the Handshake Protocol, the Alert Protocol, the Change Cipher Spec Protocol, and the Application Data Protocol. TLS Handshake Protocol allows the endpoints to authenticate each other and negotiate encryption algorithms and symmetric encryption keys before the application starts transmitting data. The endpoints can be authenticated by using public key cryptography (e.g., RSA or DSA) or with shared secrets (TLS-PSK). The TLS Alert Protocol is used for informing the other endpoint of errors or warnings and the TLS Change Cipher Spec Protocol is used for changing ciphering parameters. The TLS Application Data Protocol refers to the content type that is used for the data of the application that is used on top of TLS. [18]

TLS provides very different connection properties than EAP because it provides a reliable connection that can carry high data volumes as opposed to EAP which is designed to support only small amounts of payload data. IF-T is used on top of TLS as an application which means that IF-T is encapsulated within the TLS Record Protocol using application data content type. [26]

The IF-T binding to TLS allows for either the TNC Client or TNC Server to establish the TLS session and initiate the assessment process. Examples of situations where either the TNC Client or TNC Server might initiate the assessment include:

- The TNC Server notices suspicious behavior on an endpoint
- The TNC Server receives new security policies that require reassessment of the TNC Client
- The TNC Client notices a change in its local security posture
- The TNC Client wants access to a protected network resource

Because either endpoint may initiate the TLS session, it means that both the TNC Client and the TNC server need to be listening to a TCP port that is known by the other party. When the TNC Server is the initiator, the TNC Client is effectively acting as the TLS server which means that the TNC Client should also possess an X.509 certificate that is used to protect the initial portion of the TLS handshake. Provisioning X.509 certificates to all TNC Clients requires a quite heavy Public Key Infrastructure (PKI) which is one of the reasons that it is actually recommended for the TNC Client not to listen for connection requests coming from the TNC Server. Instead, the TNC Client should proactively establish and maintain a TLS session to the TNC Server so that either party may initiate the assessment process over the existing TLS session. [26]

The IF-T over TLS message exchange occurs in three phases: TLS Setup, IF-T Pre-Negotiation and IF-T Data Transport. During the TLS Setup phase the TLS session initiator (usually the TNC Client) establishes a TCP connection between the NAR and NAA. After the TCP connection is established the TLS Handshake Protocol is used for negotiating the cryptographic settings for the TLS session. During the negotiation the TLS session responder (usually the TNC Server) provides a trustworthy X.509 certificate that is used for authenticating the TLS responder. The TLS initiator may also provide a client certificate to authenticate to the TLS responder. If the client supports X.509 certificate with Subject Key Attestation Evidence (SKAE) [28] extensions it can be used in conjunction with a Trusted Platform Module (TPM) [27] to provide hardware-based platform authentication. Using TNC with TPM is described in section 2.7. After the TLS handshake the NAR and NAA have a protected session to exchange messages which allows the protocol to transition to the IF-T Pre-Negotiation phase. [26]

The IF-T Pre-Negotiation phase is performed only once during the lifetime of a TLS session. This phase is used for the NAR and NAA to discover each other's IF-T capabilities and establish a context that will be kept intact during the lifetime of the TLS session. The context is basically the version of the IF-T protocol to be used. If client side authentication has not been performed during the TLS handshake the NAA can authenticate the client during the IF-T Pre-Negotiation phase using IF-T client authentication messages (see table 3. Finally the IF-T session transitions into IF-T Data Transport phase, where it remains for the lifetime of the session. [26]

During the Data Transport phase IF-TNCCS messages are exchanged encapsulated in IF-T messages. Because of the full-duplex nature of the underlying TLS session, either NAA or NAR may start transmitting an IF-TNCCS message received from its upper layer component (TNC Client or TNC Server). The initiator encapsulates the IF-TNCCS message in an IF-T message, assigns a message identifier to the message, and forwards it to the

Table 3: IF-T Message Types [26]

Message Type	Contents
IFT_TYPE_EXPERIMENT	Reserved for experimental use.
IFT_VERSION_REQUEST	A version negotiation request including the versions supported by the sender.
IFT_VERSION_RESPONSE	Contains the IF-T version selected by the responder.
IFT_CLIENT_AUTH_REQUEST	Contains the authentication methods that the TNC Server supports and includes a request to choose an authentication method supported by the TNC Client.
IFT_CLIENT_AUTH_SELECTION	Contains the authentication method chosen by the TNC Client.
IFT_CLIENT_AUTH_CHALLENGE	Contains the authentication challenge sent by the TNC Server to the TNC Client.
IFT_CLIENT_AUTH_RESPONSE	Contains the client's response to the authentication challenge.
IFT_CLIENT_AUTH_SUCCESS	Indicates that the client was authenticated successfully.
IFT_TNCCS_20_BATCH	Contains an IF-TNCCS 2.0 message.
IFT_TNCCS_SOH_10_BATCH	Contains an IF-TNCCS 1.0 message.
IFT_TNCCS_XML_10_BATCH	Contains an XML-based IF-TNCCS message.
IFT_ERROR	Contains an IF-T Error Message.
IFT_NON_TNC_DATA	Contains non-TNC standard application data. This can be used by applications that share the same TLS session but do not want to define a vendor-specific message type to identify the message.

responder. The responder decouples the IF-TNCCS message and forwards it to its upper layer component. [26]

Figure 14 depicts the three phases of an IF-T session and the message exchange that occurs within each phase.

2.5.5 Policy Enforcement Point Interface (IF-PEP)

IF-PEP protocol is used for sending network access decisions from the NAA to the PEP as depicted in Figure 15. This occurs after the endpoint integrity has been assessed and the NAA has decided the level of access granted to the AR. The access decision triggers an action on the PEP, either to grant full access for the endpoint or to isolate the endpoint. There are two possible types of isolation: no access and limited access. The limited access can be used to provide remediation services for the endpoint. [29]

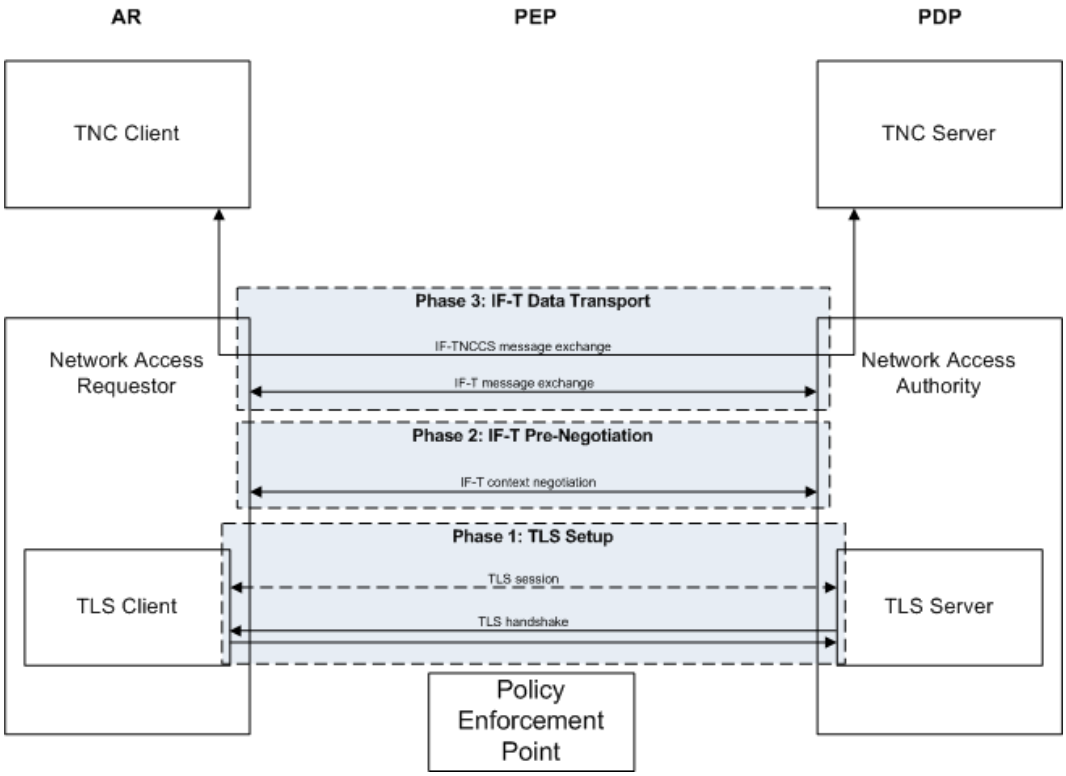

Figure 14: IF-T session phases

IF-PEP defines three isolation methods: binary-based, VLAN-based and filter-based isolation. Binary-based isolation is the simplest isolation technique in which the endpoint is either granted full access or it is completely blocked from the network. Binary-based isolation is straightforward to implement but it doesn't support remediation of non-compliant endpoints. In Local Area Networks (LANs) that support IEEE 802.1Q [30] and 802.1D [31], virtual LANs (VLANs) can be used to provide isolation. VLANs allow network operators to divide physical Ethernet-based network into multiple logical networks and it is a technique originally used for improving network performance and scalability. Later it has been widely adopted also as a network security mechanism providing higher level of security between network segments within a physical network. In 802.1Q each Ethernet frame is tagged with a VLAN identifier, or tag, which identifies the logical network in which the frame belongs to. In VLAN-based isolation The PEP (an 802.1Q-capable switch) assigns tag for the endpoint's traffic based on the network access decision made by the NAA. This allows defining different levels of access. For example, non-compliant endpoints may be isolated into a logical network (VLAN) that contains only necessary services to provide remediation for the endpoint. [29]

In filter-based isolation the network traffic of the AR is filtered by the PEP using Access Control Lists (ACLs). The ACLs consist of Access Control Entities (ACEs) each of which defines either a permit or deny rule for a service. The PEP evaluates an incoming packet against the ACLs to either allow or block the packet. In filter-based isolation the ACLs are generated based on the integrity state of the endpoint. This allows a granular way of

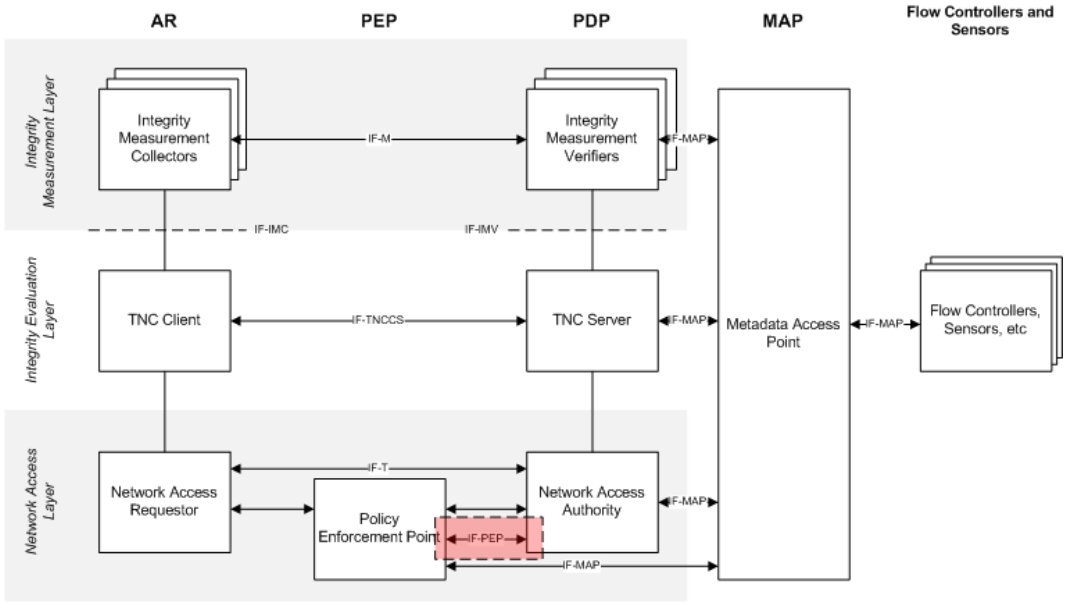

Figure 15: The IF-PEP

isolating the endpoint as it can be allowed to communicate only with specific IP addresses and TCP ports. It allows also isolation from other isolated endpoints even if they reside on the same IP subnet, which is not possible to achieve using VLAN-based isolation. [29]

Using RADIUS as IF-PEP protocol is depicted in Figure 16. Implementation of the isolation techniques described earlier is quite straightforward using RADIUS. Binary-based isolation is natively supported by RADIUS by using the ACCESS-ACCEPT and ACCESS-REJECT messages [16]. The NAA responds to the PEP with an ACCESS-ACCEPT message when the endpoint should be given access, and with an ACCESS-REJECT when the endpoint should be blocked. For example, if the outcome of the assessment is that the AR is not compliant with the integrity requirements the NAA may prevent it accessing the network by responding to the PEP with an ACCESS-REJECT message. And similarly, if the AR is compliant the NAA responds with an ACCESS-ACCEPT message. [29]

There are a few options to provide VLAN-based isolation using RADIUS depending whether tagged or untagged VLANs are being used. When tagged VLANs are used the logical networks are differentiated using 802.1Q tags which allows multiple network interfaces (i.e., switch port) to belong to multiple VLANs. Untagged VLANs are physically separated which means that each port belongs to only one untagged VLAN. RADIUS VLAN attributes [32] can be used for providing VLAN-based isolation for both tagged and untagged VLANs. Isolation for untagged VLANs can be also implemented using RADIUS tunnel attributes [34] as described in [33]. [29]

Filter-based isolation can also be implemented using existing attributes within RADIUS protocol. RADIUS specification [16] contains a Filter-Id attribute that the NAA can use to provide named filters for the PEP.

Handshake Retry is an important function of the remediation process in the TNC architec-

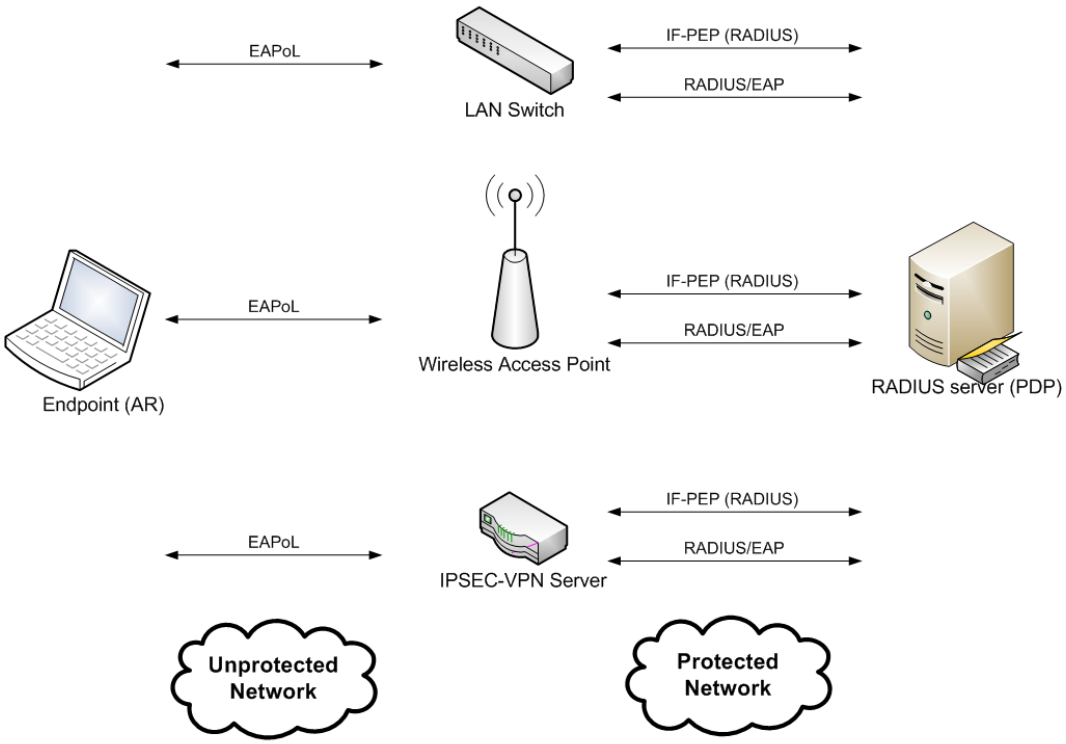

Figure 16: Network-Based PEP in TNC Architecture

ture. When the endpoint has remediated itself after first being isolated, Handshake Retry allows it to get re-evaluated against the security policy. RADIUS has been extended with dynamic authorization extensions that include two messages needed to implement Handshake Retry: Disconnect and Change-of-Authorization (CoA) [35]. At any time the NAA may send a Disconnect message to the PEP to cause immediate termination of the session, or to update the access policy by sending a CoA message including the necessary VLAN attributes. [29]

2.5.6 Metadata Access Protocol (IF-MAP)

A Metadata Access Point (MAP) is a TNC element that stores state information about users, devices and flows in a network. This information includes client address bindings and integrity and authentication status. MAP Clients (MAPCs) may publish information to a MAP Server, search information on a MAP Server and subscribe for notifications when the information on the MAP Server changes. So a single MAP Client may publish, search and subscribe. Often, however, a MAP Client is either a publisher or subscriber. For example, a TNC Server publishes integrity status information about the clients to a MAP Server, and a Flow Controller (e.g., a firewall) subscribe to this information so that they can use it to allow or block traffic flows by clients. If the integrity state of a client changes, the TNCS updates this information to the MAP Server. The MAP Server notifies the Flow Controllers of the compliancy change so that and the Flow Controller

can update its access rules. In this example the TNC Server and the Flow Controller are MAP Clients. The TNC Server is a publisher, and the Flow Controller is a subscriber. The relation between MAP Clients and MAP Server is depicted in Figure 17. [36]

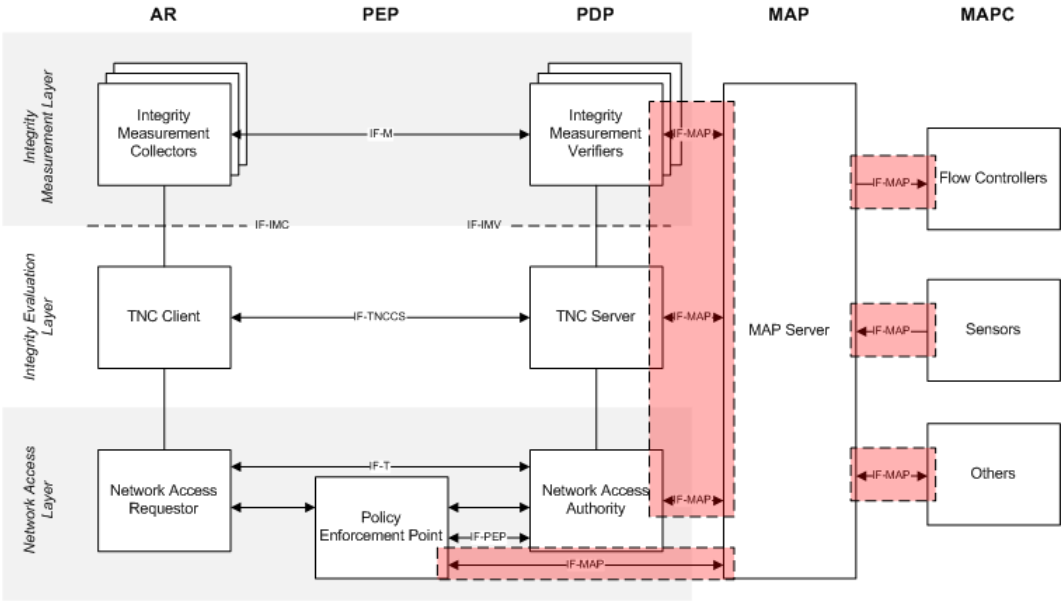

Figure 17: The IF-MAP

IF-MAP is the protocol used between MAP Clients and MAP Servers. IF-MAP supports the following use cases:

- A PDP publishes AR authentication, VLAN and other status information to a MAP.
- A TNC Server publishes integrity status of an AR to a MAP.
- A DHCP server publishes address and lease information associated with an AR.
- A Flow Controller detects a traffic flow from an AR and queries a MAP Server to find out the integrity and authentication status of the AR to decide whether to allow or block the traffic flow.
- A Flow Controller subscribes to notifications from a MAP Server about changes in client state information so that it can make appropriate enforcement adjustments to existing flows.
- A Sensor (e.g., an Intrusion Detection System) publishes information related to an AR (e.g., vulnerability detection) to a MAP Server.
- A PDP queries a MAP Server for metadata that a MAP Client has associated with an AR (e.g., vulnerability information). The PDP uses the metadata to make the access decisions. The PDP subscribes for notifications about changes in AR metadata so that it can adjust the access decisions as necessary. [36]

All IF-MAP operations and data types are represented using Extensible Markup Language (XML). IF-MAP contains two types of data: identifier and metadata. Identifier is a single, globally unique value within a range of values defined in the IF-MAP XML schema. For example, the ip-address identifier type schema element defines an identifier range consisting of all possible IP addresses. Relationships between identifiers are described using links. IF-MAP defines link as a bi-directional binding between two identifiers. For example, a DHCP server might create a link between a MAC address identifier and an IP address identifier. Table 4 lists the standard identifiers of IF-MAP. [36]

Table 4: IF-MAP Standard Identifiers [36]

Identifier	**Description**
AccessRequestType	A request for access to a network by an endpoint.
DeviceType	A physical or virtual device attempting to access the network.
IdentityType	An end-user accessing the network.
IPAddressType	IPv4 or IPv6 address
MACAddressType	Ethernet MAC address

IF-MAP metadata is represented as typed values which are defined in the schema. Metadata is always associated with a particular identifier or link. There are two types of metadata: standard and vendor-specific. Standard metadata is defined in [36] and vendor-specific metadata is used when the standard is not applicable. Table 5 lists the standard metadata types. [36]

Table 5: IF-MAP Standard Metadata types [36]

Metadata type	**Description**
access-request-device	Associates a device with an access request.
access-request-ip	Associates an access request with an IP address.
access-request-mac	Associates an access request with a MAC address.
authentication-as	Associates an end-user with a specific access request.
authenticated-by	Connects an access-request with an ip-address of a PDP or other authentication server.
capability	Refers to a collection of privileges assigned to an endpoint.
device-attribute	Associates a customizable attribute with a device.
event	Refers to some activity of interest detected on the network (e.g., p2p traffic).
layer2-information	Describes layer 2 information (e.g., VLAN) related to a specific access request.
ip-mac	Links an IP address to a MAC address.
role	Refers to a collection of privileges assigned to a specific access-request and identity.

Figure 18 illustrates an example of relationships between IF-MAP identifiers and metadata. The ovals represent identifiers and the lines depict the links between them. The rect-

angles represent the metadata elements that can be related to either links or identifiers. As the picture shows some of the metadata types (e.g., authenticated-as) are so-called simple metadata types that do not have any content whereas others can contain multiple data elements (e.g., layer2-information).

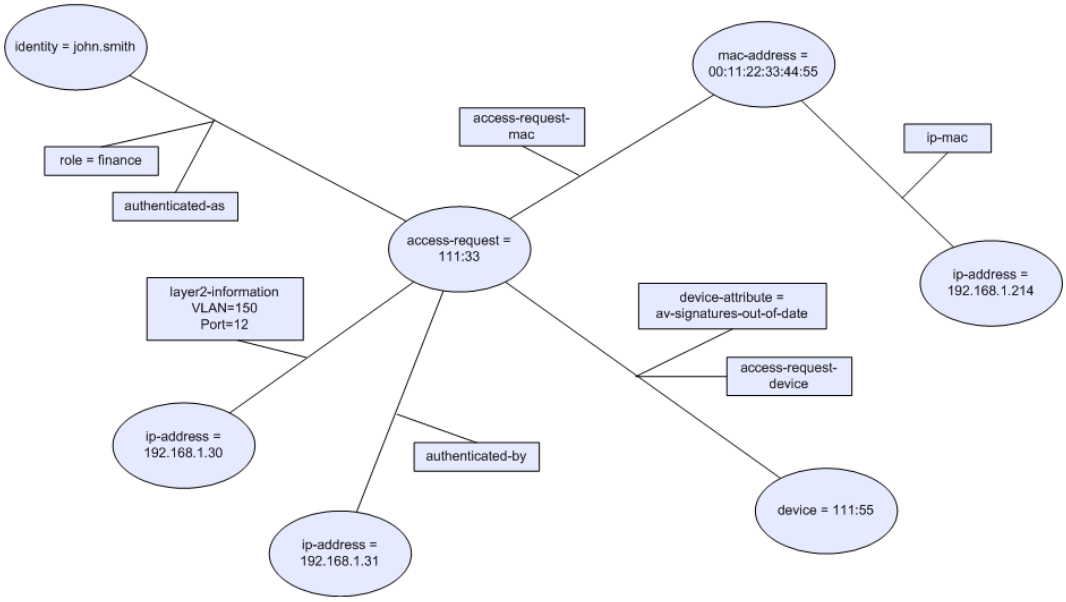

Figure 18: An example of IF-MAP metadata and identifier relationships [36]

IF-MAP is a half-duplex protocol. The message exchange between a MAP Client and a MAP Server involves the client sending a request to the server, and the server sending a response to the client. There are five types of request messages defined in IF-MAP: publish, search, subscribe, poll and purgePublisher. Publish requests are used for creating, updating and deleting metadata associated with one or more identifiers or links. Publish requests are for publishing metadata only as links and identifiers are conceptually never created or destroyed. [36]

Search requests are used for retrieving metadata from the MAP Server. The MAP Server responds with an XML document comprised of resulting identifiers and links along with the metadata associated with them. A MAP Client can send a subscribe request to subscribe for notifications when the metadata on the MAP Server is updated. Poll requests are used by the MAP Clients to request notification of metadata updates based on the client's subscription. Finally purgePublisher is used for removing all metadata associated with a particular publisher, typically used by a MAP Client to purge its own data after a system reset. [36]

Because IF-MAP is essentially a protocol to transport XML documents it is very convenient to use Simple Object Access Protocol (SOAP) [37] for exchanging IF-MAP messages. SOAP is an XML based protocol for exchanging structured information. SOAP usually operates on top of other application layer protocols, most notably Hypertext

Transfer Protocol (HTTP). IF-MAP binding for SOAP is defined in [36].

2.6 Phases in TNC

So far we have examined the different components and interfaces that comprise the TNC architecture. This section describes on a higher level the three phases that form the network access process in TNC: assessment, remediation and isolation.

During the assessment phase the IMVs validate the integrity measurements provided by the IMCs. The integrity measurements are evaluated against the security policies defined by the network operator, and based on this evaluation the IMVs can make one of three recommendations: allow, isolate or block. In addition the IMVs may send remediation instructions to the IMCs. Isolation is the phase where due to shortcomings in the integrity requirements the AR is redirected into a remediation network where it can apply the necessary changes and updates to comply with the policies. There are a number of technologies that can be used to provide isolation but currently the TNC architecture supports VLAN containment and IP filters. After the AR is isolated it has to apply the necessary changes into configurations, install missing security updates, etc. in order to become compliant with the policies and gain access to the network. The process of applying the changes is called remediation. [3]

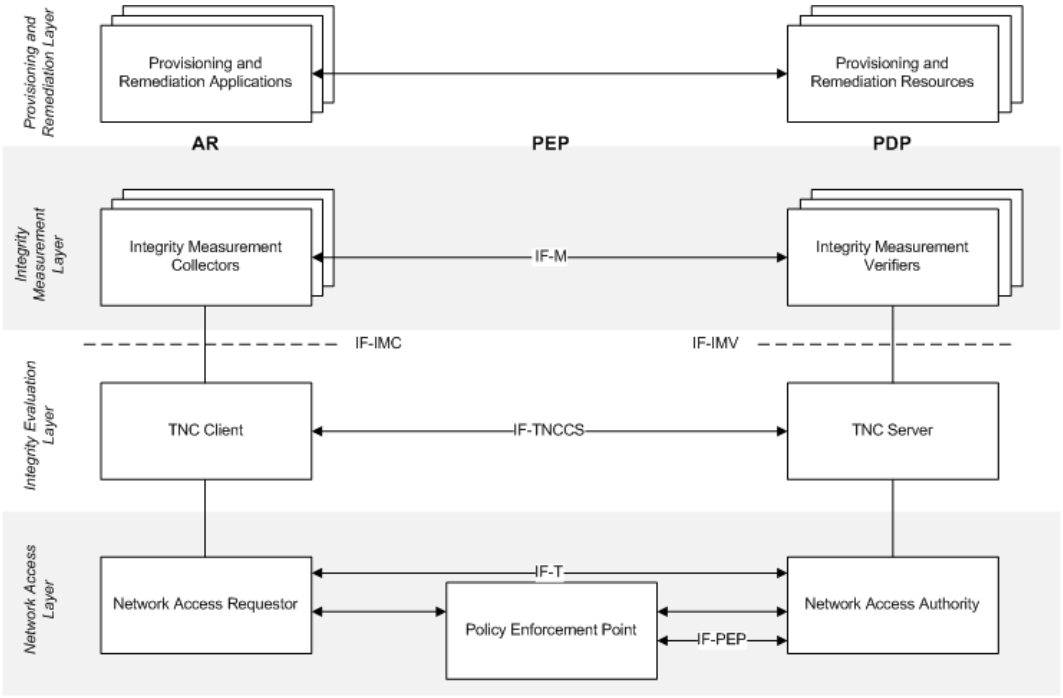

Figure 19: The Provisioning and Remediation Layer in TNC [3]

To provide support for remediation an additional layer of components called Provisioning and Remediation layer is needed. This layer and its relation to the rest of the TNC architecture is depicted in Figure 19. The Provisioning and Remediation layer contains all

the necessary applications and services to provide remediation. Provisioning and Remediation Application (PRA) is an entity that communicates with the IMCs and provides it with specific types of integrity information. An example of a PRA could be update agent software that obtains the latest security updates for an operating system. The PRA could also be implemented as a part of an IMC. [3]

The second entity within the Provisioning and Remediation layer is the Provisioning and Remediation Resource (PRR). PRR is a service that provides resources for updating the AR into compliance. Using the previous example of the update agent software the PRR would be an update server where the PRA could obtain the security updates. [3]

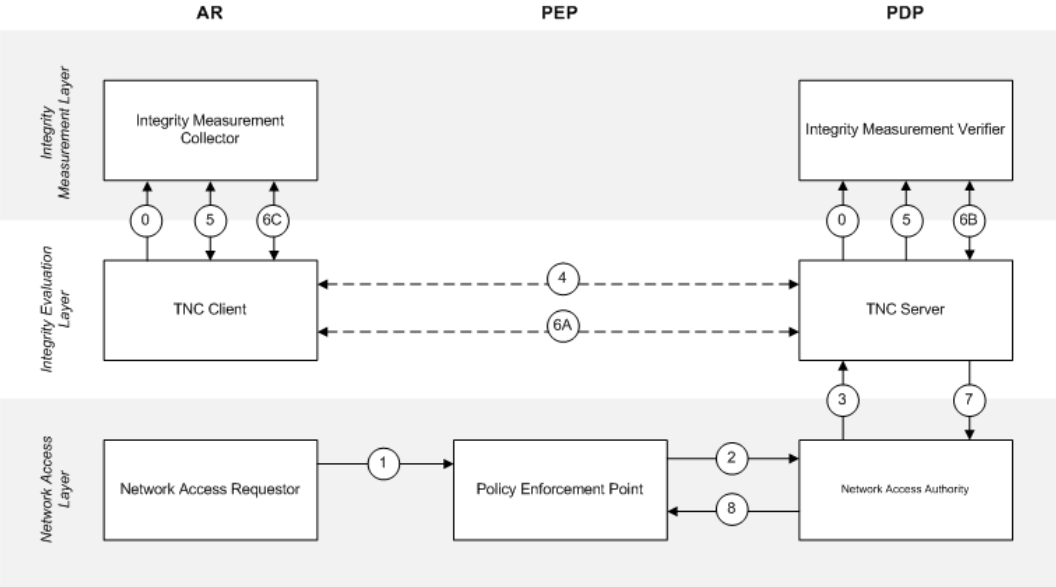

Figure 20: The TNC Message Flows [3]

Figure 20 presents an example message flow occurring during a TNC access control dialog. This example presents only a basic scenario of a TNC message exchange assuming that no remediation occurs. The first message flows occur already prior establishing a network connection. The TNCC and TNCS must load and initialize the relevant IMCs and IMVs that are used to measure the integrity state of the endpoint. The NAR initializes the network access request by establishing a network connection with the PEP. The PEP sends a network access decision request to the NAA which may perform user authentication. Assuming that the authentication succeeds the NAA forwards the connection request to the TNCS. The TNCS may perform (mutual) Platform Credential Authentication (flow 4) with the TNCC to verify that the endpoints are trusted by each other. Assuming that the authentication succeeds the TNCS informs the IMVs of a new connection request and similarly the TNCC informs the IMCs that a new Integrity Check Handshake is going to be carried out. After that the TNCC and TNCS start exchanging messages pertaining to the integrity handshake (flow 6A). The IMCs send the messages containing integrity information to the TNCC which forwards the messages to the TNCS and subsequently to the IMVs. The IMVs validate the measurements and either respond to the IMCs or provide a recommendation for the TNCS (flow 6B). Finally after the Integrity Check Handshake

has completed the TNCS sends its TNCS Action Recommendation to the NAA which consequently indicates its access decision to the PEP. [3]

2.7 TNC and the Trusted Platform Module

In addition to the interfaces and protocols described earlier the TNC architecture extends to the usage of hardware protected identity and integrity information. This extension allows a device to communicate platform proof-of-identity (Platform Credential Authentication) and platform integrity information (Integrity Check Handshake) as part of an authentication process to an authentication server (PDP). The chain of trust is built bottom-up, and the basis is the Trusted Platform Module (TPM) [27] hardware bound to the device. [3]

Trusted Platform Module is a security device that allows storing cryptographic keys and provides such functionalities as remote attestation and sealed storage. Remote attestation can be used for detecting changes on the hardware and software configuration of the platform. This is achieved by generating a certificate from the current hardware and software configuration. Sealed storage can be used for protecting private information by binding it to the platform. This means that the information can be obtained by using the same combination of hardware and software configurations. This is a feature utilized by hard drive encryption software to allow only the trusted platform to decrypt the data on the hard drive. At the time of manufacture a random encryption key called endorsement key (EK) is embedded within the TPM chip. The EK is a public/private key pair that can be used to identify the platform. The private portion of the key is never released from the TPM. [27]

Figure 21 illustrates how the TNC architecture can be extended with Platform Authentication using the TPM. The TPM provides four functionalities that are needed for the Platform Authentication: protected capabilities, integrity measurement and storage, integrity reporting and attestations. Protected capabilities refer to a set of commands that have exclusive access to shielded locations (e.g., a protected memory space) where it is safe to process sensitive data. Platform Configuration Registers (PCRs) on the TPM can be used to store integrity measurements and optionally cryptographic keys used for authentication during the Integrity Check Handshake. [3]

After the integrity measurement process the measurements are stored in an integrity log and digests of the measurements are stored in PCRs to detect any corruption of the data. The measurements are then combined into an integrity report which is signed with a private key and sent to the verifier. Finally, attestation is the process certifying for the accuracy of information, so that the relying party can use the attestation to decide whether it trusts the platform. [3]

The Platform Authentication introduces a new layer on the TNC architecture: Platform Trust Services (PTS) as depicted in Figure 21. PTS provides an abstraction layer for the TNC Client and IMCs that access the protected capabilities within the TPM. The TPM is accessed using the TCG Software Stack (TSS) [38] which is a standard Application Programming Interace (API) for accessing the functions of the TPM. The PTS functionality

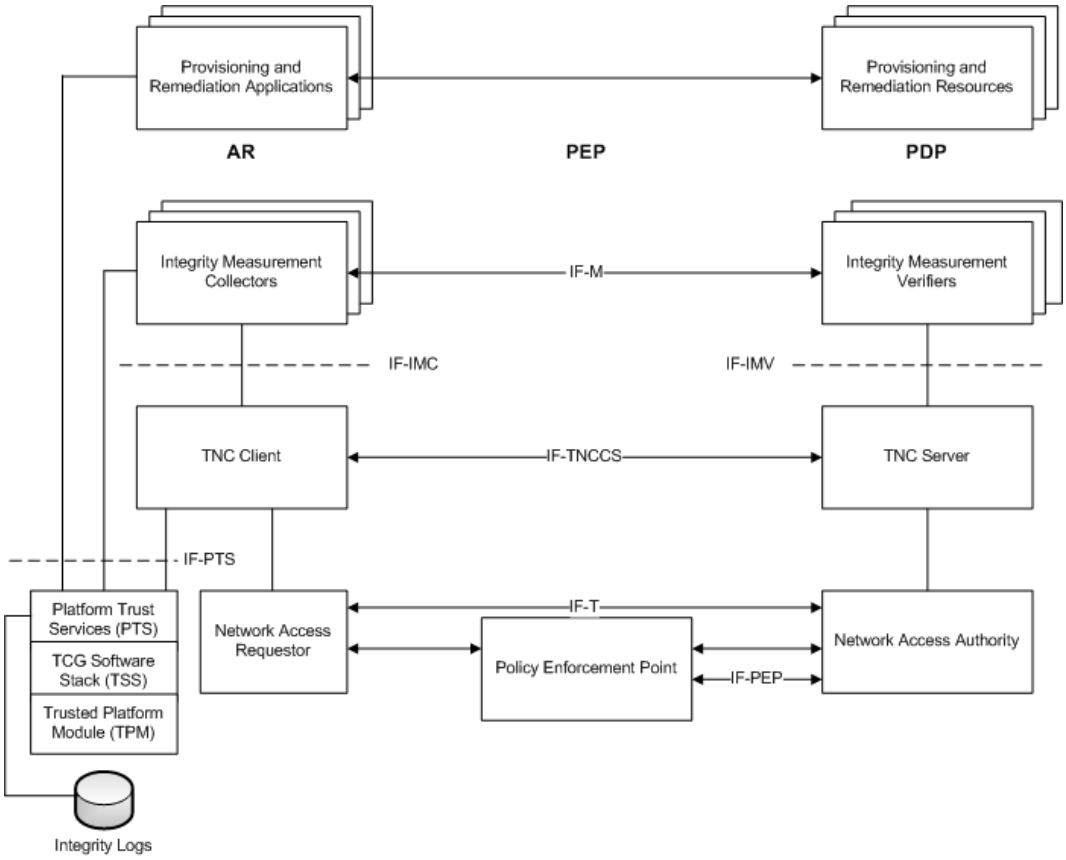

Figure 21: The TNC architecture with the Trusted Platform Module [3]

is provided to upper layer components using Platform Trust Services Interface (IF-PTS). IF-PTS is not yet standardized but the TNC-WG is planning to standardize it in the future. [3]

Platform ownership is an important concept in context of TNC and Platform Authentication. A common situation is that the PDP and PEP is owned by the same party (e.g., network operator or IT department). The AR may or may not have the same owner but in context of Platform Authentication this ownership should be re-evaluated. If the owners are different both parties have to trust to a third party called Privacy Certificate Authority (Privacy CA) in order to successfully carry out Platform Authentication and remote attestation. The Privacy CA issues Attestation Identity Key (AIK) certificate for trusted platforms. The trusted platforms are authenticated using the endorsement key of the TPM. If the PDP trusts the same Privacy CA the AR can use the AIK to authenticate to the PDP. [3]

2.8 Federated TNC

So far we have presented an architecture that enables network operators to enforce security policies regarding endpoint posture. However, this model assumes that the endpoint

and the TNC Server belong to the same security domain, i.e. have a direct trust relationship with each other. Federated TNC extends the integrity verification to cover multiple organizations or security domains. It defines a new protocol called Federated TNC protocol (IF-FTNC) which enables communication of IF-M attributes, IF-TNCCS Action Recommendations and IF-MAP metadata between security domains that have a trust relationship with each other. When two or more security domains share a trust relationship they are referred as federated. The role of federated TNC is depicted in Figure 22. [39].

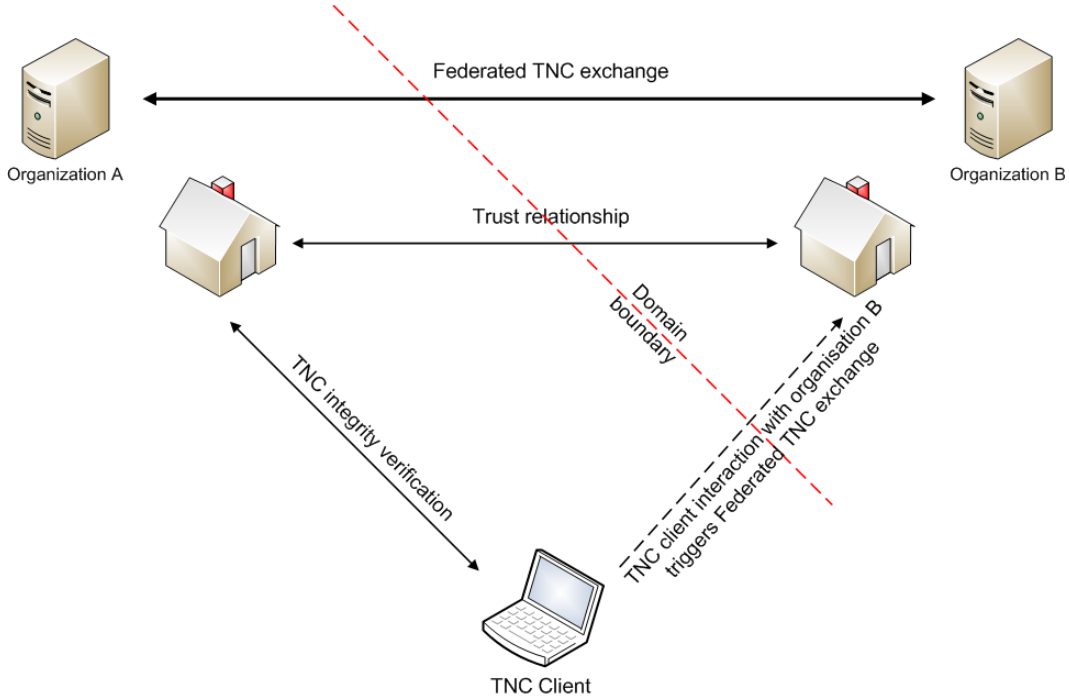

Figure 22: Federated TNC [39]

The conceptual model of Federated TNC distinguishes three actors: the endpoint (TNC Client), the Asserting Security Domain (ASD) and the Relying Security Domain (RSD). ASD is the domain that has knowledge of the endpoint's Security Posture Information (SPI). RSD is the domain where the endpoint is requesting access to. When the endpoint requests access at the RSD the RSD requests the endpoint's SPI from the ASD, which in turn assesses the endpoint's posture. The three parties and their relationships with each other is depicted in Figure 23. [39]

Currently the IF-FTNC defines two supported use case profiles: Roaming Assessment Profile and Web Assessment Profile. The Roaming Assessment Profile enables organizations with roaming agreements to use integrity-based authorization for roaming clients. It allows the host network (the RSD) to request the endpoint's SPI from the endpoint's home network (the ASD). The use of Roaming Assessment Profile is illustrated in Figure 24. [39]

The TNC is often considered as a network access control mechanism used when the endpoint is connecting the network. However after the network connection is successfully established the endpoint may interact with network web resources that aren't within the

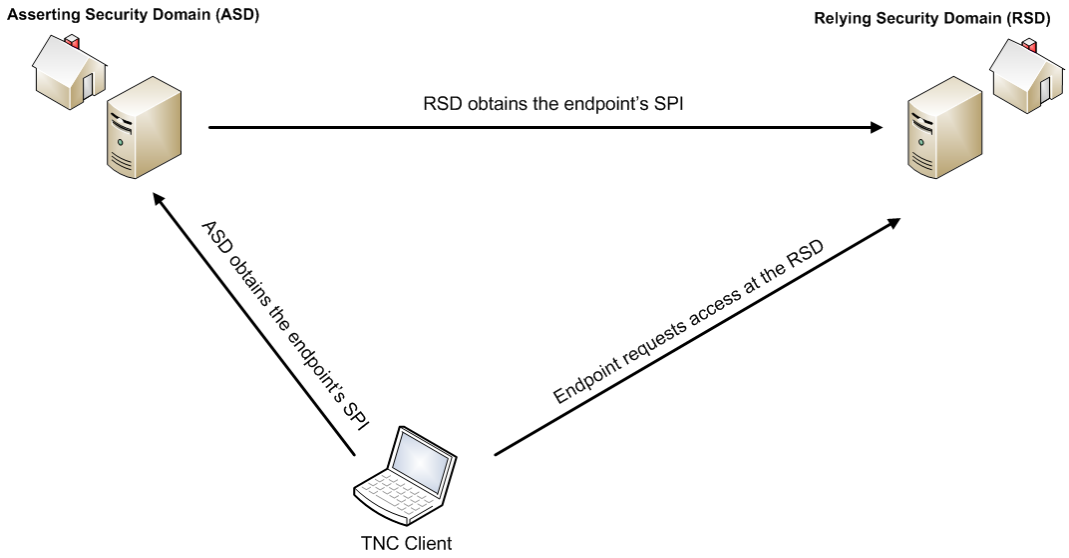

Figure 23: The parties in the Federated TNC [39]

same security domain. The service providers for these web resources (e.g. partner organizations or software-as-a-service providers) may also be interested of the client's security posture. For example, a partner organization might be concerned that if the client's credential information will be compromised if the client's security posture is not solid. Using Web Assessment Profile the web resource (the RSD) may obtain the client's SPI from the client's home network (the ASD) as illustrated in Figure 25. [39]

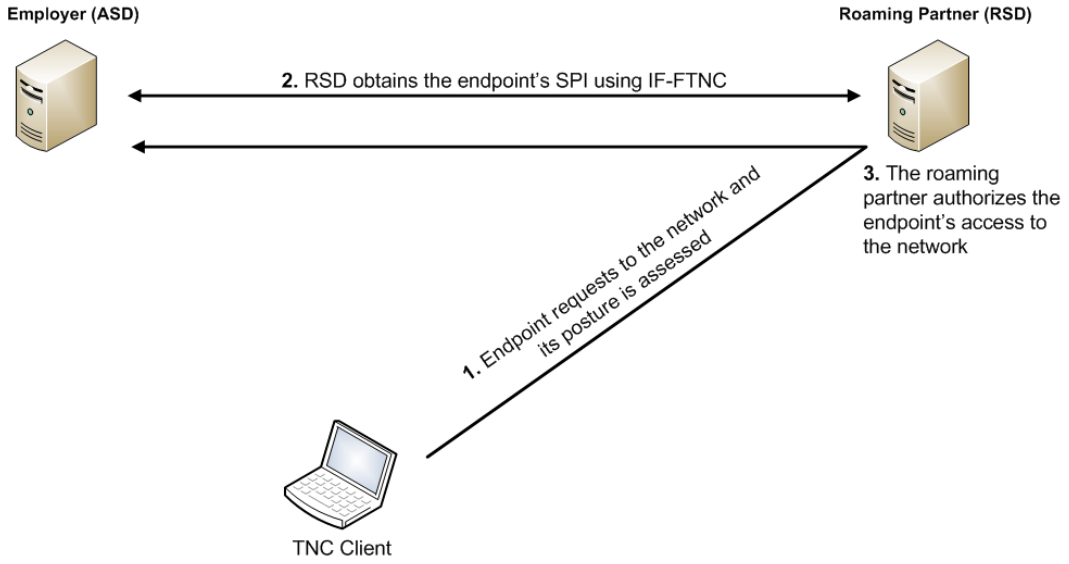

Figure 24: The Roaming Assessment Profile [39]

The IF-FTNC protocol is built on Security Assertion Markup Language (SAML) which is an XML-based framework for exchanging authentication and authorization information between security domains. SAML operates between three key parties: the user, the Iden-

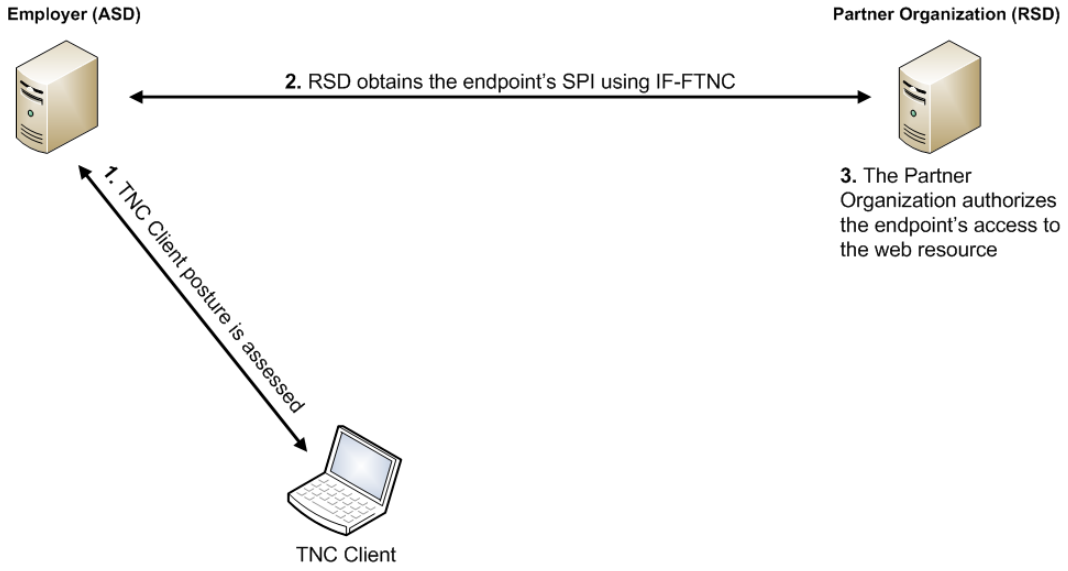

Figure 25: The Web Assessment Profile [39]

tity Provider (IdP) and the Service Provider (SP). Identity Provider is the party holding the identity information about the users. Service Provider owns the service or data that is being accessed by the user. SAML specification defines XML-based assertions, protocols, bindings and profiles. SAML assertion is an XML document that contains security information about a user. In clmProtocol defines how SAML elements are encapsulated within request and response elements. Bindings define how the requests and responses are transferred using existing protocols. Currently there is only one binding which is SAML over SOAP. Finally, profiles define how SAML assertions, protocols and bindings combine to support a defined use case. In IF-FTNC the ASD operates a TNCS which performs authentication and posture assessment functions, as well as a SAML attribute authority which can respond to queries for SAML assertions. The RSD operates a SAML Service Provider and requester, which obtains assertions from the attribute authority. [40] [39]

2.9 Summary

This chapter described the architecture of Trusted Network Connect which is a framework that provides endpoint integrity verification as a network access control mechanism. The TNC architecture is comprised of entities, components and interfaces. The fundamental entities in the architecture are the Access Requestor (AR), Policy Enforcement Point (PEP) and Policy Decision Point (PDP). AR is the client seeking access to the network. PDP makes the access decision based on the integrity information provided by the AR during the assessment process. PEP enforces the access decision made by the PDP by granting or restricting access to the network. The authorization process is comprised of three phases: assessment, isolation and remediation. During the assessment phase the PDP verifies the integrity of the AR. Isolation is the process of restricting access of non-

compliant clients, and during remediation the non-compliant clients are brought compliant. TNC utilizes existing protocols for communication between the components. These protocols include Extensible Authentication Protocol (EAP), Transport Layer Security (TLS), Remote Authentication Dial In User Service (RADIUS) and IEEE 802.1X.

In addition to the basic entities the TNC also supports Metadata Access Point (MAP) and Flow Controllers and Sensors for storing metadata of the clients in the network and using this information to protect the network. TNC also integrates with Trusted Platform Module (TPM) to provide hardware-based authentication and integrity protection, and provides integrity assessment over multiple security domains using federated TNC.

3 Commercial implementations of NAC-EI

This chapter describes in detail Microsoft Network Access Protection (NAP) as a commercial implementation of NAC-EI. At first we describe the architecture behind the NAP and in the last section we compare NAP with TNC and try to find out similarities and differences between the open standard architecture and the commercial implementation.

3.1 Introduction

Network Access Protection (NAP) is a collection of operating system components to provide system health state validation as a basis for network access decisions. NAP allows the network operator to define security policies to determine whether a client accessing the network is compliant. Non-compliant clients can be provided with limited access to the network until the health policy requirements have been met. NAP has been an integral part of Windows operating systems since Windows Server 2008 and Windows Vista. Also Windows XP Service Pack 3 supports NAP. [41]

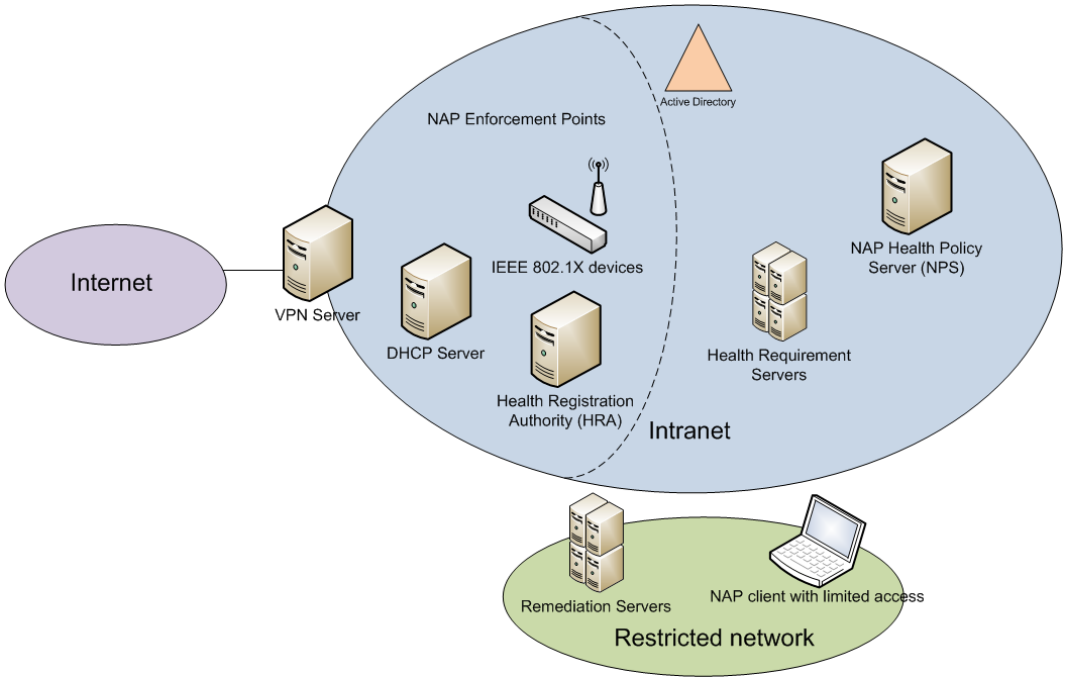

Figure 26: NAP Components [41]

NAP architecture provides the following functionalities to support network access control based on endpoint health state: health state validation, network access limitation, automatic remediation and ongoing compliance. Health state validation is the process of determining whether the client computer is compliant with the health policy requirements. Network access limitation provides limited access to the network for non-compliant clients. Automatic remediation enables non-compliant clients to become compliant without user

intervention. And finally, ongoing compliance is the process of automatically updating clients so that they remain compliant even if the health policy requirements change. [41]

3.2 NAP Components

Figure 26 describes the different components within the NAP architecture. NAP Client is the client computer seeking access to the network. NAP Health Policy Servers store health requirement policies and provide health state validation services. The component in Windows Server 2008 acting as NAP Health Policy Server is Network Policy Server (NPS). NPS can also act as an AAA server. The actual identity information is stored in Active Directory (AD) which is a directory service providing authentication and authorization services. [41]

Health Policy Servers communicate with Health Requirement Servers to obtain latest health requirements. For example, a Health Requirement Server for an anti-virus program would probably keep track on the latest anti-virus signature files. The actual enforcement of NAP health requirement policies is done by NAP Enforcement Points. Figure 26 presents some examples of NAP enforcement including DHCP server, IEEE 802.1X devices and VPN server. NAP Enforcement Points use the help of NAP Health Policy Server to evaluate the health state of the clients. Non-compliant clients are restricted into a restricted network containing remediation servers that provide necessary services and resources to bring non-compliant clients into compliant. Compliant clients are provided with a certificate by Health Registration Authority (HRA) which obtains the certificates from a Certification Authority (CA). [41]

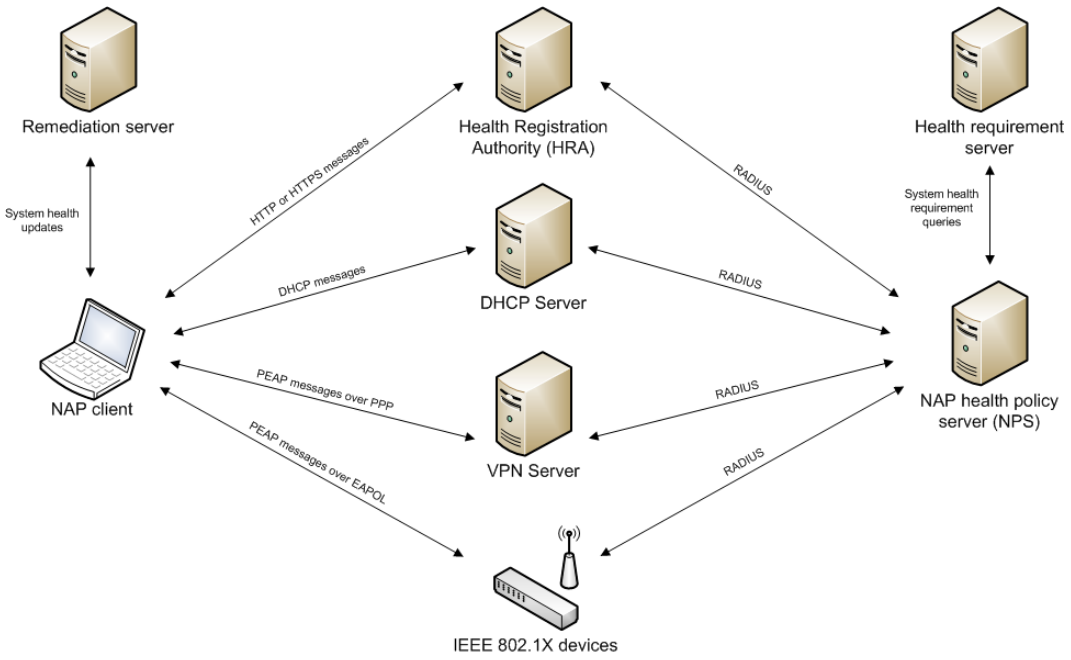

Figure 27: Communcation between NAP components [41]

3.3 NAP Client Architecture

Figure 28 describes the different components within a NAP Client. NAP Client is a computer running a Windows operating system supporting NAP functionality. The NAP Client platform consists of three kinds of components: System Health Agent (SHA) components, NAP Enforcement Client (EC) components and a NAP Agent component. SHA is a component that maintains and reports some piece of system health information. Examples of System Health Agents include software components that check the status of anti-virus signatures or operating system updates. Windows operating system provides SHAs for measuring the system health on a high level but vendors are expected to develop SHAs for measuring and reporting vendor-specific health information. SHAs can also have a corresponding remediation server. Using the previous examples, the SHAs could communicate with remediation servers providing latest anti-virus signatures or operating system updates. [41]

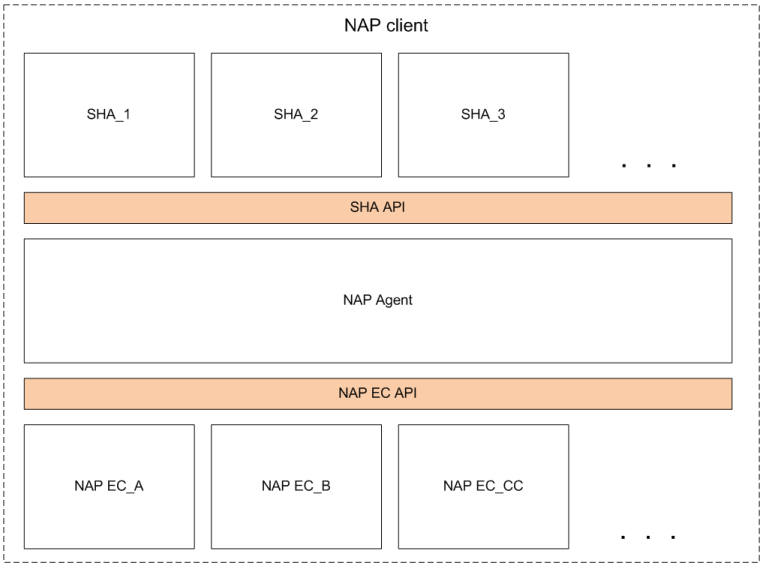

Figure 28: NAP client architecture [41]

The NAP Client uses NAP Enforcement Clients to communicate with NAP Enforcement Points. Each network access technology should have its own NAP EC, e.g., one NAP EC to communicate with DHCP servers, another to communicate with IEEE 802.1X devices. Third-party vendors may provide their own NAP ECs. [41]

NAP Agent is the component that maintains the system health information of the NAP Client and facilitates communication between NAP ECs and SHAs. NAP Agent communicates with the SHAs and NAP ECs using SHA API and NAP EC API. SHA API is an interface that allows SHAs to register with the NAP Agent, to provide system health status, to respond to requests for system health status and for the NAP Agent to pass remediation instructions to the SHAs. Similarly, the NAP EC API allows the NAP ECs to register with the NAP Agent and to pass information received from the NAP Enforcement Point to the NAP Agent. [41]

To report the health status an SHA generates a Statement of Health (SoH) and passes it to the NAP Agent. An SoH can contain one or more elements of system health information. Each time an SHA updates its status it generates a new SoH and forwards it to the NAP Agent. The NAP Agent aggregates all the SoH elements into a System Statement of Health (SSoH) which represents the overall health status of the NAP Client. The NAP Agent passes the SSoH to a NAP EC which forwards it to the NAP Enforcement Point. [41]

3.4 NAP Server Architecture

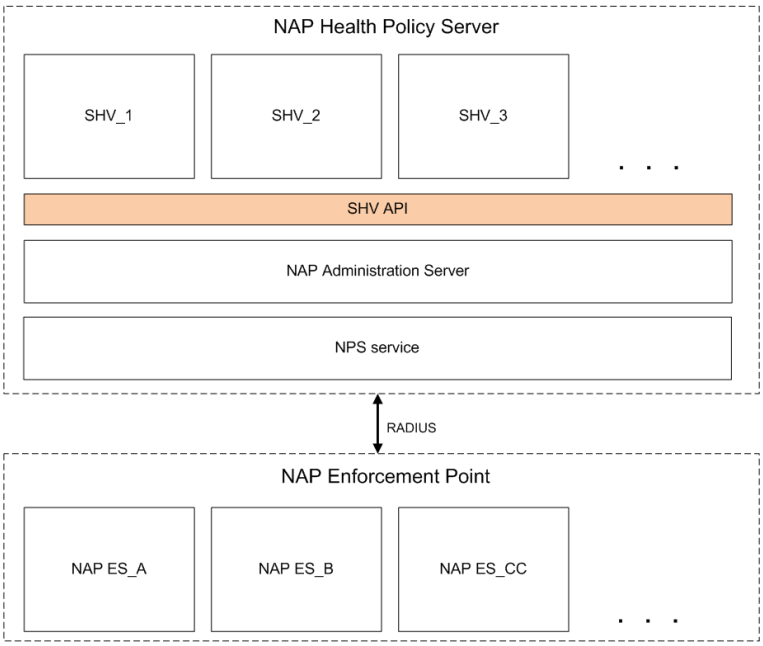

Figure 29: NAP Server architecture [41]

As illustrated in Figure 29 the NAP server-side architecture consists of two entities: NAP Health Policy Server and NAP Enforcement Point. NAP Health Policy Server consists of three kinds of components: System Health Validators (SHVs), a NAP Administration Server and an NPS service. SHV is a component that is responsible of validating the integrity of a specific piece of system information. Each SHV communicates with a corresponding SHA on the client side. An SHV may also communicate with one or more Health Requirement Servers to obtain system health requirements. However this is not mandatory. An example of an SHV that doesn't need to communicate with a Health Requirement Server is an SHV that checks that a client firewall software is enabled on the NAP Client. [41]

NPS Service is an implementation of RADIUS server that receives the RADIUS Access-Request message, extracts the SSoH sent by the NAP Client and forwards it to NAP Administration Server. The NAP Administration Server is the component that facilitates communication between the NPS Service and the SHVs as illustrated in Figure 30. It re-

ceives the SSoH from the NPS Service, disaggregates the SoHs from the SSoH and passes the SoHs to the corresponding SHVs. When an SHV receives a Statement of Health (SoH) from the NAP Administration Server it compares the system health status in the SoH with the required system health state. After the validation the SHV generates a Statement of Health Response (SoHR) and passes it to the NAP Administration Server. The SoHR indicates whether the health requirements were met by the SoH and optionally remediation instructions. For example, the SoHR might include the IP address of a remediation server holding the latest anti-virus signature files. The NAP Administration Server in turn generates a System Statement of Health Response (SSoHR) by combining all the SoHs received from the SHVs. [41]

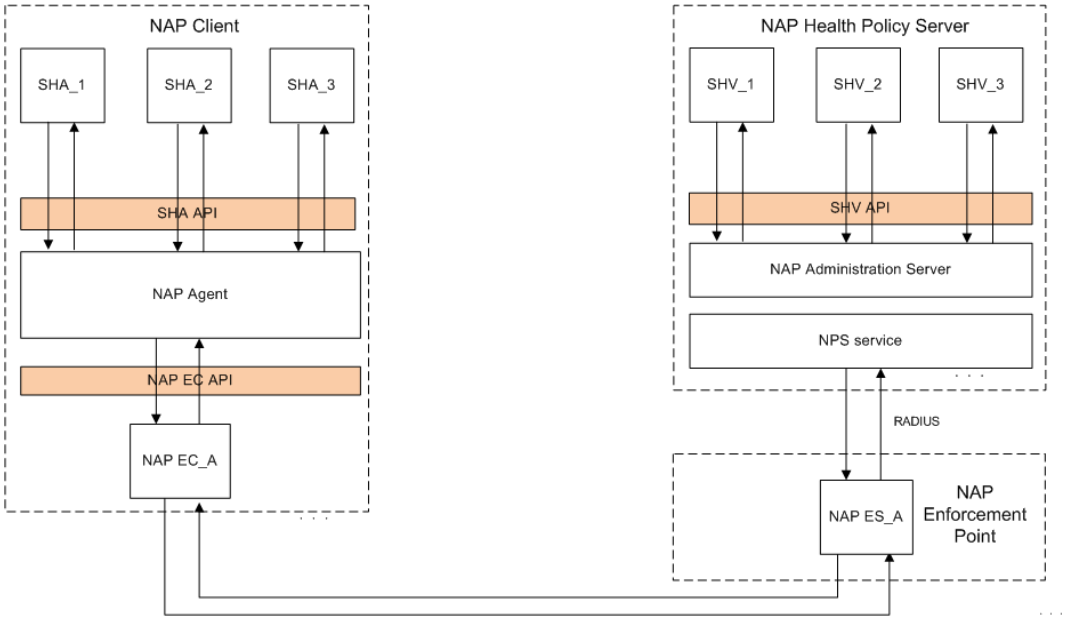

Figure 30: Message exchange between NAP Client and Server components [41]

NAP Enforcement Point is the entity that enforces the network access decisions made by the Health Policy Server. The Enforcement Point contains one or more NAP Enforcement Server (NAP ES) components, each of which implements some network access technology. Windows Server 2008 provides three built-in NAP ESs: NAP ES for IPsec communications, NAP ES for DHCP and NAP ES for Terminal Server (TS) Gateway connections. [41]

3.5 NAP Enforcement Methods

The NAP network access process can be divided into four phases: system health status reporting, network policy compliance verification, network access limitation and automatic remediation. During system health status reporting the NAP Client provides an SSoH for the Health Policy Server. During the network policy compliance verification the Health Policy Server validates the SSoH against the health policy requirements. Network access limitation is the process of isolating a noncompliant client into a restricted network, and

bringing the client compliant without user interaction is called automatic remediation. This section describes how these functionalities can be provided using IPsec-protected communication, 801.1X-based authorization, VPN connections and DHCP addressing. [41]

3.5.1 IPsec Enforcement

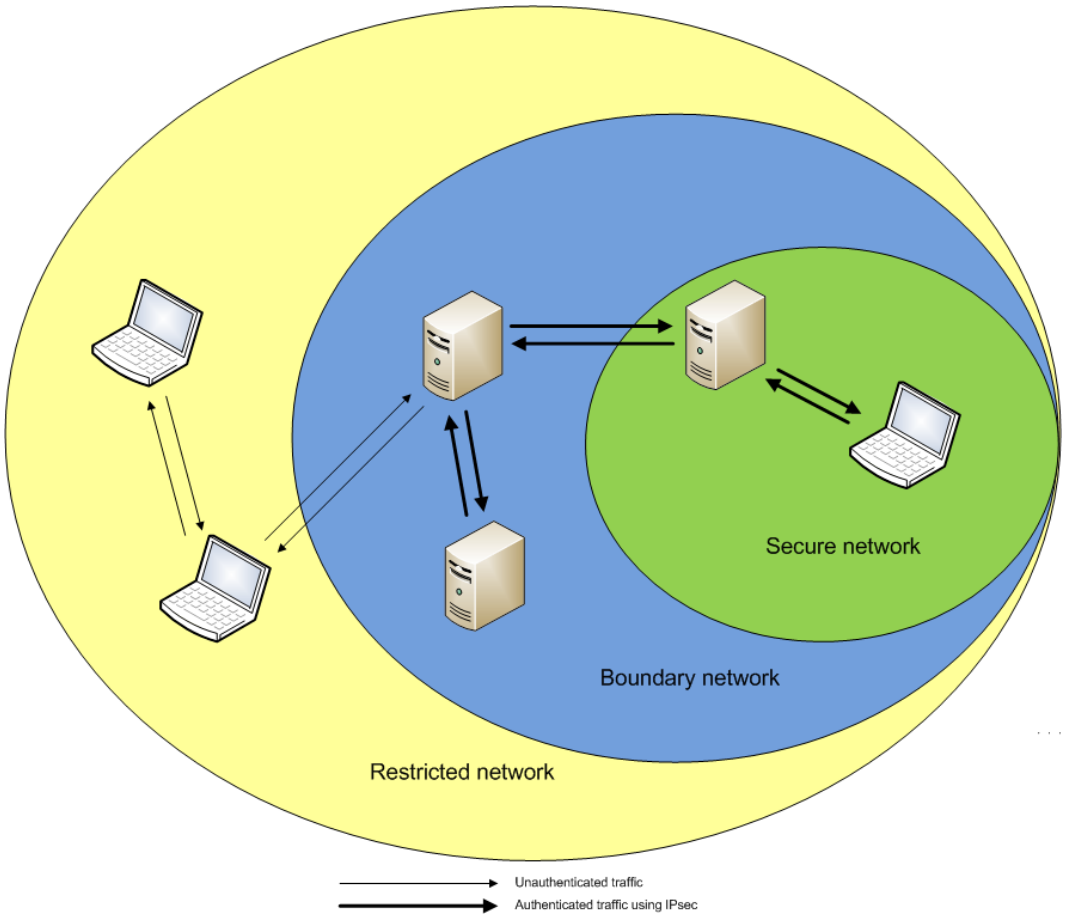

Figure 31: NAP IPsec enforcement [41]

NAP can be used in conjunction with IP Security Architecture (IPsec) to provide NAP enforcement. As opposed to VPN or 802.1X enforcement the IPsec enforcement is implemented by each individual network node, rather than at the edge of the network. The enforcement is based on the health certificates that are assigned for each compliant client during the compliance assessment. Using IPsec allows the enforcement to be limited for specific clients on a subnet or specific TCP or UDP ports assuming that there is an infrastructure in place to distribute IPsec policies (e.g., Active Directory Group Policies). [41]

IPsec enforcement divides the network into three logical partitions: secure network, boundary network and restricted network. This classification is illustrated in Figure 31. The

secure network contains the clients that are classified as healthy, i.e. possess health certificates. All clients in secure network require the use of health certificates for authenticating incoming IPsec traffic. They may communicate to clients in all networks but will accept only authenticated incoming connections from the secure network and the boundary network. [42]

The boundary network contains the computers that hold health certificates but do not require the use of health certificates for IPsec authentication. Clients in the boundary network require authentication for connections initiated from secure network or boundary network. Connections initiated from the restricted network do not require authentication. The boundary network typically consists of the HRA and NAP Remediation Servers. This is because the HRA and the NAP Remediation Servers need to communicate with both compliant and non-compliant clients. [42]

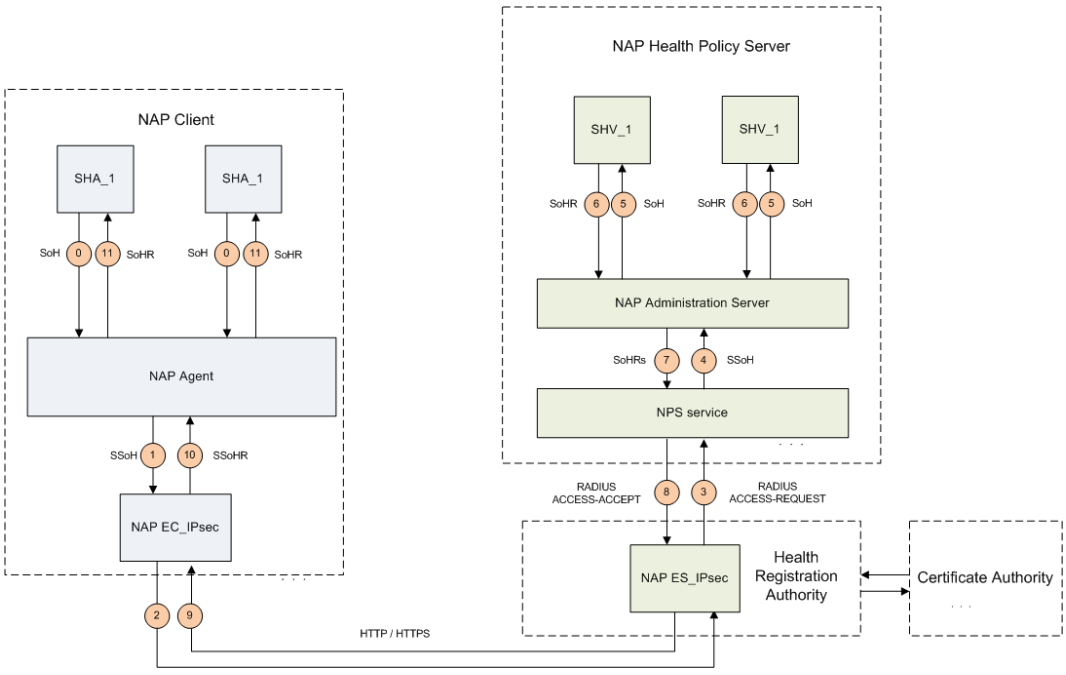

Figure 32: NAP IPsec message flow

Figure 32 describes the message flow that occurs when a NAP Client obtains a health certificate. The process begins when the client connects to the network and obtains an IP address configuration. At this point the client does not have a health certificate so it is placed in the restricted network. It can communicate only with computers within the restricted and boundary networks. The NAP Agent obtains the SoHs from the SHAs. This can actually occur before the client connects to the network which is the reason that it is described with message flow 0 in the figure. The NAP Agent combines the SoHs into a System Statement of Health (SSoH) and forwards it to the NAP EC. The NAP EC initiates an HTTP or HTTPS session with the HRA and uses this session to pass the SSoH to the HRA. The HRA extracts the SSoH from the HTTP or HTTPS message and passes it to the NPS service as a RADIUS Vendor-Specific Attribute (VSA) in a RADIUS Access-Request message. The NPS service on the NAP Health Policy Server receives the Access-

Request message, extracts the SSoH from the RADIUS VSA and forwards it to the NAP Administration Server. The NAP Administration Server segregates the SoHs from the SSoH and passes each SoH to the corresponding SHV. The SHVs evaluate the SoHs against the system health requirements and generate SoHRs on the basis of this evaluation. The NAP Administration Server receives the SoHRs from the SHVs and passes them to the NPS service. The NPS service evaluates the SoHRs against the configured set of health policy requirements and generates the SSoHR. The NPS service encapsulates the SSoHR into a RADIUS Access-Accept message which indicates whether the client gets restricted or unrestricted access to the network. The HRA extracts the SSoHR from the RADIUS message and passes it to the NAP EC using HTTP or HTTPS. The NAP EC forwards the SSoHR to the NAP Agent which forwards the individual SoHRs to the SHAs. If the NAP Client is compliant the HRA obtains a health certificate from the Certificate Authority (CA) and issues it to the client. The client stores the certificate into its local certificate store and configures its IPsec policies to use this certificate to authenticate IPsec connections. At this point the client is located in the secure network. If the client is not compliant with the health policy requirements the SHAs perform the necessary remediation actions to bring the client compliant. After the remediation is completed the SHAs send the updated SoHs to the NAP Agent which initiates a new assessment process identical to the one depicted in Figure 32. This process continues iteratively until the client is evaluated as compliant and issued a health certificate. [42]

3.5.2 IEEE 802.1X Enforcement

In IEEE 802.1X enforcement an 802.1X-capable device is acting as NAP Enforcement Point. The isolation is based on either VLAN tagging or IP filtering. In VLAN tagging the client traffic is distinguished into separate Virtual Local Area Networks (VLANs) using VLAN IDs. In IP filtering the 802.1X-capable device applies IP filters to the client traffic allowing only access to resources according to the Access Control List (ACL) defined in the filter. [41]

Figure 33 illustrates the message flow between the different NAP client and server components when IEEE 802.1X enforcement is used. The assessment process starts when either the 802.1X client or the 802.1X-capable network device (e.g., switch) initiates 802.1X authentication using EAP over LAN (EAPOL) protocol. During the authentication the NPS service, which is acting as RADIUS authentication server, sends an EAP-Request/Identity message to the EAPHost NAP EC which responds with an EAP-Response/Identity message containing the user or computer name of the client. After that the NPS service sends an EAP-Request/Start PEAP message to initiate a Protected EAP (PEAP) conversation. PEAP is an authentication protocol developed for environments where physical security of the network cannot be guaranteed (e.g., wireless networks) and it works by initially negotiating a TLS session and then conducting EAP conversation within it [43]. [41]

The NAP Client and Health Policy Server negotiate a protected TLS session. The message exchange occurs logically between the NPS service and the NAP EC while the 802.1X-capable network device is acting as a pass-through device. The message exchange between the NPS service and the 802.1X-compliant device is conducted using RADIUS

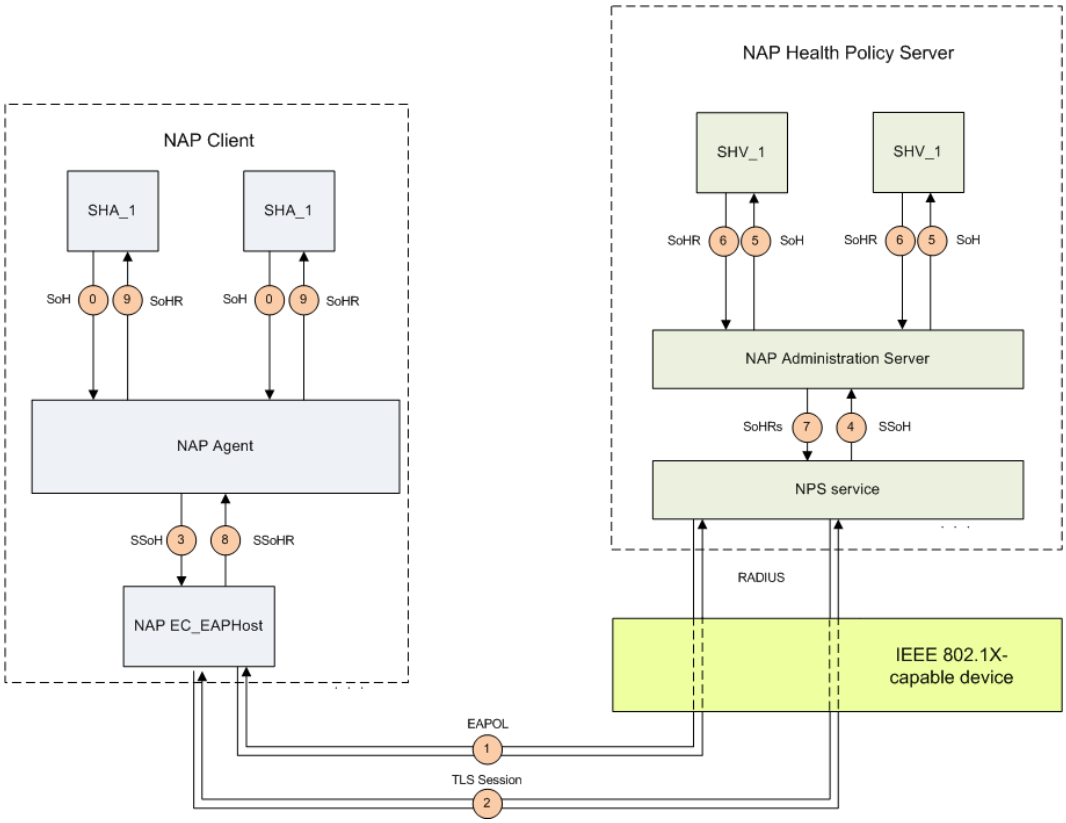

Figure 33: NAP message flow with IEEE 802.1X enforcement

Access-Request, Access-Accept and Access-Challenge messages. After the TLS session has been established the NPS service sends a PEAP-TLV message containing a request for the SSoH of the NAP Client. The NAP EC queries the NAP Agent for the SSoH and the NAP Agent responses with the SSoH aggregated from the SoHs received from the SHAs. The NAP EC forwards the SSoH to the NPS service. At this point the NPS service sends a request to the NAP Client to authenticate itself using either computer or user credentials. The authentication is performed using one the PEAP authentication methods, e.g., PEAP-Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2). After the authentication the NPS service extracts the SSoH from the PEAP-TLV message and passes it to the NAP Administration Server which then forwards each SoH to the corresponding SHVs. The SHVs validate the SoHs and respond with SoHR generated based on the validation. The NAP Administration Server passes the SoHRs to the NPS service which compares the SoHRs against the defined set of health policy requirements and based on this comparison generates a SSoHR. The NPS service sends a PEAP-TLV message containing the SSoHR to the NAP EC and a RADIUS Access-Accept message to the 802.1X device. If the NAP Client is considered non-compliant the Access-Accept message contains an access profile to limit the traffic from the NAP Client to the restricted network. If the client is considered compliant the message does not contain an access profile and the client will be provided unrestricted access to the network. The NAP EC extracts the SSoHR from the PEAP-TLV message, passes it to NAP Agent which in turn

forwards each SoHR to the corresponding SHAs. Each SHA analyze the contents of the SoHRs and, if necessary, perform the necessary remediation steps. After the remediation the SHAs send the updated SoHs to the NAP Agent. The NAP Agent generates an updated SSoH and passes it to the NAP EC which initiates a new 802.1X authentication and system health validation. [41]

3.5.3 DHCP Enforcement

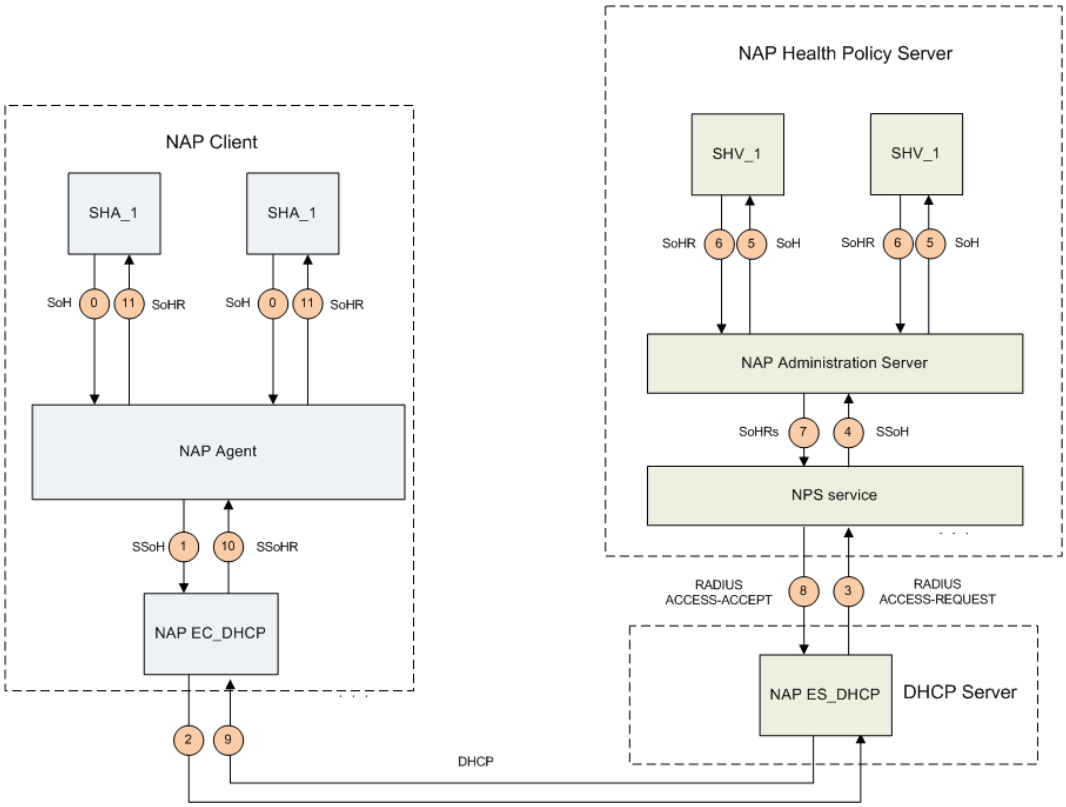

Figure 34: NAP message flow with DHCP configuration

Dynamic Host Configuration Protocol (DHCP) is a network protocol used for delivering host-specific network configuration information on a TCP/IP network. DHCP uses a client-server where client is the host seeking for configuration parameters and server is the host that allocates network addresses and delivers configuration parameters to clients. The parameters contain such information as the addresses of the Domain Name System (DNS) servers, and the default gateway and subnet mask of the network [44]. The DHCP configuration initiates when the client connects to the network and sends a DHCPDiscover message which is a broadcast message sent to discover the available DHCP servers on the network. The DHCP server responds to the client with a DHCPOffer message that contains the configuration parameters configured on the server. After the client has received the offer it sends a DHCPRequest to request for an address lease from the DHCP server. Finally, the server sends a DHCPAck to confirm the address lease. [45]

DHCP can be used to provide NAP enforcement. In this case the DHCP server is acting as NAP Enforcement Point. The DHCP enforcement works so that the DHCP server provides valid network configuration parameters only to clients that meet the health policy requirements. Non-compliant clients are provided with DHCP Router value 0.0.0.0 which means that the clients do not have a default gateway. Traffic to same subnet is restricted by assigning subnet mask of 255.255.255.255 which means that the client routes all traffic to the default gateway, which in this case is not configured. To enable access to remediation servers the DHCP server utilizes the Static Route Option [44] to provide routes in the restricted network. There are a couple of limitations with DHCP enforcement. The first limitation is that it is restricted to IPv4, so it cannot be used with IPv6. The second limitation is that even though the client is provided with limited network configuration parameters, a malicious user with administrative permissions on the client may alter the network configuration to gain full access to network. [41]

The message flow during NAP DHCP enforcement is illustrated in Figure 34. When the NAP Client connects to the network the DHCP NAP EC within the NAP Client queries the NAP Agent for a SSoH. The NAP Agent generates a SSoH from the SoHs received from the SHAs and passes it to the DHCP NAP EC. The DHCP NAP EC creates a DHCPDiscover message containing the SSoH as a Microsoft vendor-specific DHCP option and sends it to the DHCP NAP ES component on the NAP-enabled DHCP Server. The DHCP NAP ES extracts the SSoH from the DHCPDiscover message and sends it to the NPS service within a RADIUS Access-Request message. The NPS service on the NAP Health Policy Server extracts the SSoH from the Access-Request message and forwards it to NAP Administration Server. The NAP Administration Server forwards each SoH from the SSoH to the corresponding SHVs which validate the SoHs and respond with a SoHR. The NAP Administration Server compares the SoHRs against the health policy requirements, generates a SSoHR and passes it to the NPS service. The NPS service creates a RADIUS Access-Accept message containing the SSoHR and sends it to the DHCP server. The DHCP server determines from the Access-Accept message whether the client is compliant and sends a DHCPOffer to the client containing the IP configuration parameters and the SSoHR as a vendor-specific option. The client sends a DHCPRequest message requesting the offered IP configuration parameters and the DHCP server responds with a DHCPAck. The DHCP NAP EC within the NAP Client extracts the SSoHR from the DHCPOffer and forwards it to the NAP Agent which subsequently forwards each SoHR to the corresponding SHAs. If the NAP Client is considered compliant the IP address configuration offered with the DHCPOffer and DHCPAck contains an IP address for the client, a subnet mask for the subnet to which the client belongs and a DHCP Router option with the value of the default gateway of the subnet. At this point the client has unlimited network access. [41]

If the client is not compliant, the default gateway value is set to 0.0.0.0 and the subnet mask to 255.255.255.255. In addition, the DHCPOffer and DHCPAck include the Static Route Option consisting of the addresses of the remediation servers. At this point the SHAs need to communicate with the remediation servers to become compliant. After the remediation has completed, the SHAs send the updated SoHs to the NAP Agent which aggregates them into an SSoH. The NAP Agent passes the updated SSoH to the DHCP

NAP EC which initiates a new DHCP session in order to gain unrestricted access to the network. [41]

3.5.4 VPN Enforcement

Virtual Private Network (VPN) is a logical network implemented on top of an existing network infrastructure to provide access to a private network over an insecure network, usually the Internet. VPNs are widely used in organizations to provide remote access to private resources over the Internet. Because the remote access clients are logically connected to the internal network it is necessary to be able to apply the same security policies for remote access clients as for the locally connected clients. NAP provides a routine for using a VPN server as a NAP Enforcement Point to enforce the same set of health policy requirements for the remote access clients as for the locally connected clients. [41]

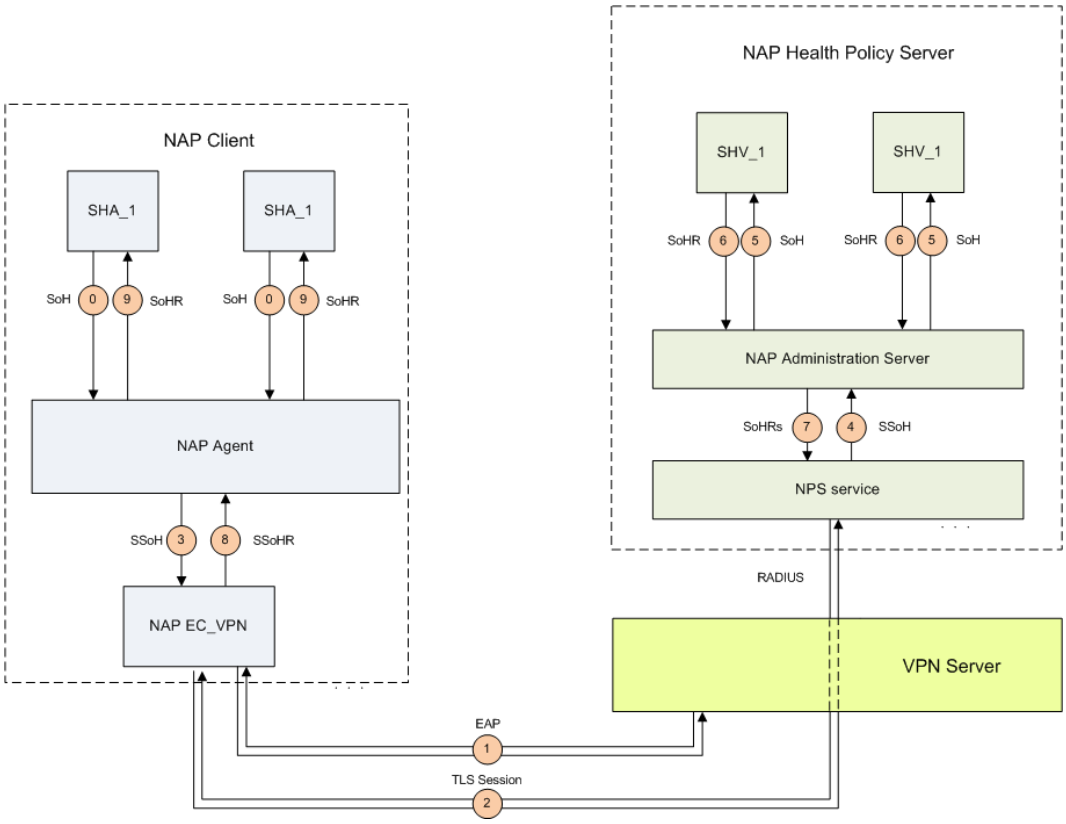

Figure 35: NAP message flow with VPN enforcement

The message flow during the VPN enforcement is described in Figure 35. The process initiates when the NAP Client establishes a VPN connection to the VPN server using one of the remote access protocols provided by the VPN server. Examples of VPN protocols include Layer Two Tunneling Protocol over IPsec (L2TP/IPsec) [46] and Point-to-Point Tunneling Protocol (PPTP) [47]. During the VPN establishment the VPN server sends an EAP-Request/Identity message to the VPN NAP EC component within the NAP Client.

The VPN NAP EC responds with an EAP-Response/Identity containing the user name of the client. The VPN server passes the EAP-Responde/Identity message to the NPS service within a RADIUS Access-Request message. After that the NPS service initiates a PEAP conversation with the VPN NAP EC using the VPN server as a pass-through device in the same manner as in the IEEE 802.1X enforcement. The NPS service sends an EAP-Request/Start PEAP message to the VPN NAP EC to establish a protected TLS session. After that the NPS service sends a request for SSoH to the NAP Client using a PEAP-TLV message. The VPN NAP EC queries the NAP Agent for the SSoH and the NAP Agent responds with a SSoH generated from the SoHs. The VPN NAP EC responds to the NPS service with a PEAP-TLV message containing the SSoH. The NPS service receives the SSoH and requests that the VPN client authenticate itself using one of the PEAP authentication methods. After the authentication is completed the NPS service extracts the SSoH from the PEAP-TLV message and forwards it to the NAP Administration Server which conveys the SoHs for the corresponding SHVs. the SHVs analyze the contents of the SoHs and respond with SoHRs. The NAP Administration Server passes the SoHRs to the NPS service which compares the SoHRs with the defined health policy requirements. The NPS service then creates an SSoHR, encapsulates it in a PEAP-TLV message and forwards it to the VPN NAP EC. The NPS service sends a RADIUS Access-Accept message to the VPN server to enforce IP packet filters. If the client is considered non-compliant the Access-Accept message contains a set of IP packet filters to limit the access of the VPN client to the restricted network. If the client is compliant, the Access-Accept message does not contain any IP filters and the VPN client will gain unrestricted access to the network. The VPN NAP EC receives the PEAP-TLV message, extracts the SSoHR from the message and passes it to the NAP Client. The NAP Client forwards each SoHR to the corresponding SHAs. If the NAP Client is not compliant with the health policy requirements the SHAs contact the remediation servers to become compliant and then create updated SoHs to initiate a new assessment process using the existing PEAP conversation. [41]

3.6 NAP and TNC

So far we have described two different architectures that support network access control based on endpoint integrity. Trusted Network Connect (TNC) is an open standard developed by Trusted Computing Group (TCG) which defines a set of vendor-neutral components and interfaces to support posture-based network access control. Network Access Protection (NAP) is a commercial product developed by Microsoft that implements similar features than the TNC. In this section we compare these two architectures with each other and try to find out differences and similarities between them. To compare these two technologies we must first take a look at the features that the architectures aim at supporting. The TNC specification identifies the following requirements [3]:

- Platform-Authentication: the verification of the identity (Platform Credential Authentication) and integrity (Integrity Check Handshake) of the platform requesting access to the network.

- Endpoint Policy Compliance: establishing a level of trust in the state of the endpoint. Endpoint Policy Compliance can be seen as an authorization mechanism in which the integrity state of the endpoint is provided as input for the authorization decision.
- Access Policy: guaranteeing that the client is authenticated and authorized before connecting to the network.
- Assessment, Isolation and Remediation: ensuring that non-compliant endpoints are quarantined from the rest of the network and possibly provided with remediation services to in order to become compliant.

Microsoft NAP provides the following features:

- Health state validation: determining whether the endpoint is compliant with the predefined set of security policies.
- Network access limitation: restricting access of non-compliant endpoints.
- Automatic remediation: the process of helping the endpoint to become compliant without user intervention.
- Ongoing compliance: updating compliant endpoints automatically to stay compliant even if the policy requirements change.

If we look at the requirements we can see that there are a lot of similarities but also some differences. The TNC requirements emphasize the importance of trust which is founded on the Trusted Platform Module (TPM). NAP architecture does not support platform authentication in the same manner. Both architectures recognize the phases of assessment, isolation and remediation. The client posture is assessed before access to the network is granted and non-compliant clients are provided with restricted access. The process of bringing a non-compliant client into compliant, i.e. remediation, is also recognized in the requirements of both architectures even though NAP emphasizes more the importance of automatic remediation without user intervention. One of the features of NAP is ongoing compliance which is not listed in the TNC requirements. However this feature is provided by the TNC IF-T protocol as described in section 2.5.4. In section 2.8 we discussed about federated TNC and IF-FTNC protocol that enables to provision TNC over multiple security domains by using trust relationships. NAP does not currently support integrity verification over multiple security domains.

If we compare TNC and NAP from a pure architectural point of view we can see that they share a lot in common. Figure 36 depicts the components and interfaces within both architectures. The blue rectangles represent the components within the TNC architecture that do not exist within the NAP architecture. The green rectangles represent the components that exist within both architectures and that can be mapped with each other. The components in the figure are named according to NAP architecture, and the names within the parentheses represent the corresponding components within the TNC.

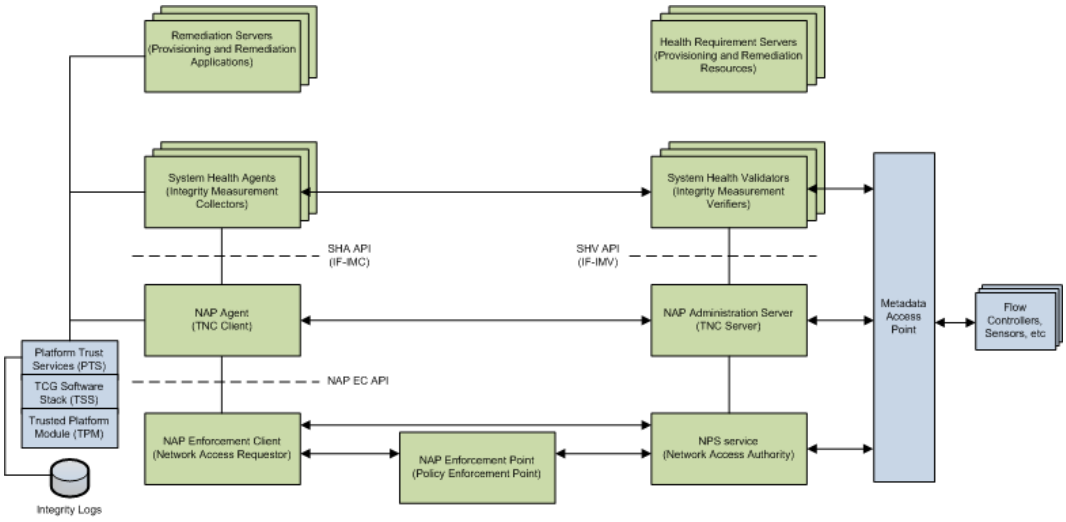

Figure 36: TNC and NAP Architectures

As we can see from Figure 36 all the components that are needed for assessment, isolation and remediation are very much alike within both of these architectures. Metadata Access Point and Flow Controllers and Sensors are used for monitoring the network and identifying changes in the network and these components are not included within the NAP architecture. The Trusted Platform Module can be used in TNC to provide hardware-based authentication and integrity verification but this feature is not available in NAP either. The SHAs perform exactly the same functionality as the IMCs. An IMC collects integrity information and sends this information to the TNC Client. Equally, an SHA creates a Statement of Health based on the gathered integrity measurements and passes it to the NAP Agent. The role of the TNC Client and NAP Agent is to aggregate the integrity measurements and exchange messages with the server side. The NAP Agent loads the SHAs using the SHA API whereas the TNC Client loads the IMCs using IF-IMC. NAP Enforcement Clients are equivalent to the Network Access Requestors in the TNC. A NAP EC establishes network connection with the NAP Enforcement Point. Just as with NARs, there may be multiple NAP ECs within the NAP Client to support different kinds of network access mechanisms. NAP Enforcement Point is the component that applies the network access decision made by the NPS service. The equivalent in the TNC is the Policy Enforcement Point, which enforces the network access decision made by the Network Access Authority. Just like the NPS service, NAA is often considered to be a RADIUS Authentication Server. The NAA or the NPS service makes the access decision based on the information that it gets from the TNC Server or NAP Administration Server. The role of the TNC Server is to facilitate communication between the IMVs and the NAA. It loads the IMVs, aggregates the data collected from the IMVs and passes it to the NAA. The role of the NAP Administration Server is equivalent.

RADIUS protocol plays a key role in both TNC and NAP architectures. It is convenient as RADIUS provides authentication and authorization capabilities, and as we discussed, the endpoint compliance verification can be seen as an authorization mechanism where the endpoint posture is used as input for the authorization decision. In the previous section we

discovered that NAP supports currently four policy enforcement methods: IEEE 802.1X, DHCP, IPsec, VPN and DHCP. TNC supports currently IEEE 802.1X-based enforcement and VPN-based enforcement as discussed in section 2.5.4.

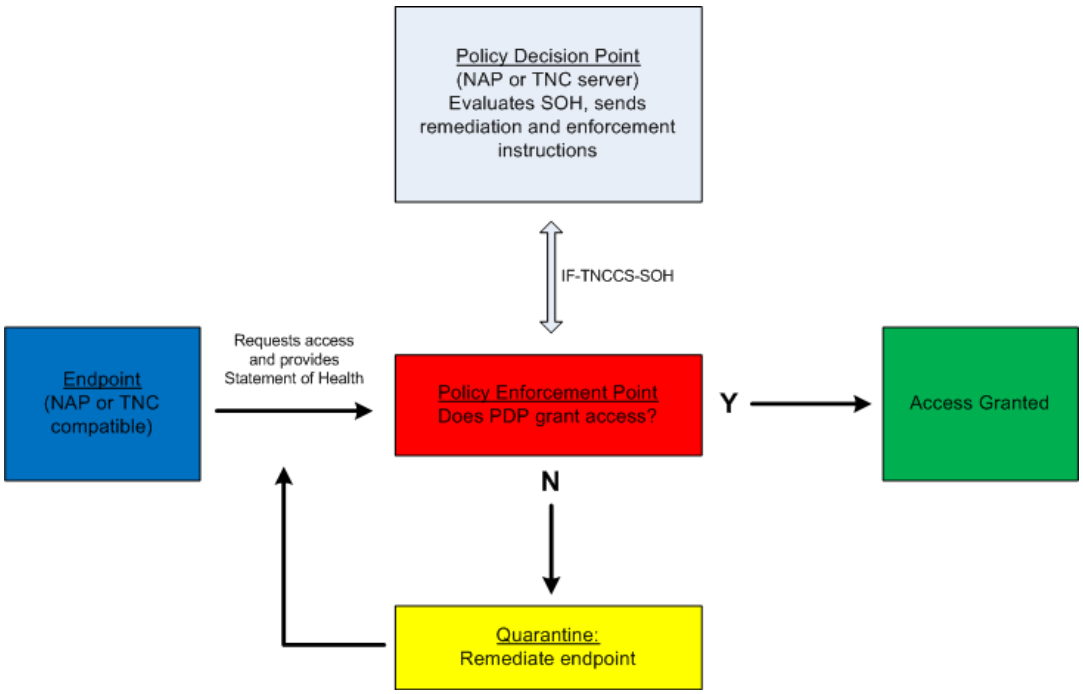

Figure 37: Interoperability between TNC and NAP [48]

As we saw in the previous paragraphs the high-level architecture of the Microsoft NAP is very similar to the TNC architecture. Of course the reason is that Microsoft has been closely involved in the TNC Work Group with other major network companies, such as Juniper. In May 2007 Microsoft and TCG announced that they will provide interoperability between TNC and NAP architectures. The first step was that Microsoft contributed its Statement of Health (SoH) protocol to the TNC Work Group. The TNC-WG published IF-TNCCS bindings for Microsoft SoH protocol (IF-TNCCS-SOH) which is an open standard protocol that provides interoperability between TNC and NAP components [13]. Figure 37 illustrates the interoperability between NAP- and TNC-compatible components. The endpoint might be a TNC Client or a NAP Client, and the Policy Decision Point might be either a TNC or NAP Server. The protocol that enables the interoperability is the IF-TNCCS-SOH which is supported by both TNC and NAP. The Policy Enforcement Point does not need to understand IF-TNCCS-SOH as it just relays the messages between the endpoint and the PDP. [48]

Figure 38 summarizes the differences and similarities between the open standard TNC architecture and Microsoft NAP architecture. The dark blue rectangle illustrates the features and functionalities of the TNC architecture whereas the light blue rectangle represents the NAP architecture. As the figure illustrates both architectures share a lot in common which enables interoperability between these architectures as discussed previously. The TNC architecture is more extensive as it provides also support for federation and monitoring

capabilities which are features not implemented in NAP. NAP implements some isolation technologies that are not supported by the TNC, at least not yet.

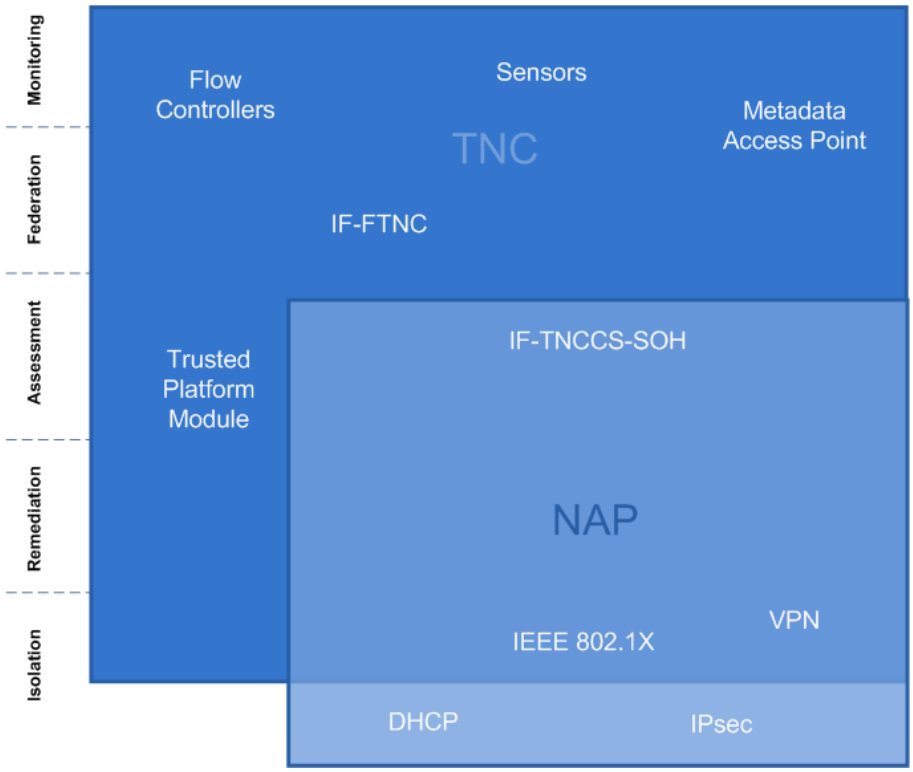

Figure 38: TNC and NAP functionalities

3.7 Summary

In this chapter we described Microsoft Network Access Protection (NAP) as a commercial implementation of NAC-EI. The three basic components within NAP architecture are the NAP Client, NAP Health Policy Server and NAP Enforcement Point. NAP Client is a computer running Windows operating system supporting NAP functionality. NAP Health Policy Server is the entity that makes the network access decision based on the integrity information provided by the client. NAP Enforcement Point either grants or restricts the access to the network based on the decision made by the Health Policy Server. The authorization process in NAP is comprised of four phases: system health status reporting, network policy compliance verification, network access limitation and automatic remediation. If the first two phases are combined into one phase, these phases are equivalent to assessment, isolation and remediation in the TNC. Currently NAP supports four enforcement methods: IPsec, DHCP, 802.1X and VPN.

4 Future of NAC-EI

In the previous chapters we described two architectures for providing integrity-based network access control: Trusted Network Connect, which is an open standard architecture, and Network Access Protection which is a commercial product implemented by Microsoft. We also compared these two architectures with each other, the features and functionalities that they provide, and the components and interfaces which build up the architectures. In this chapter we try to analyze what is the future of NAC-EI products: is it something that is going to be implemented widely in organizations, or is it just fancy architecture with no real-life use.

4.1 Status Quo

Before we can look into the future, we must first take a look at the current situation. So far we have presented only two architectures, one of which is an open standard architecture and the other a commercial implementation. In addition to that, we briefly discussed about Network Endpoint Assessment (NEA), which is an architecture developed by IETF. But in addition to these there is a wide range of products already in the market. So far we have used Microsoft NAP as an example implementation, but there are other significant software and infrastructure companies that have developed their own equivalent products. Cisco Network Admission Control (CNAC) is an infrastructure product that provides similar capabilities as NAP and TNC [49]. Unified Access Control (UAC) is an comparable product developed by network company Juniper. As Juniper is strongly contributin to the TNC-WG the UAC complies with the TNC standards. Security company Symantec also has a network access control implementation which is called Symantec Network Access Control (SNAC).

Cisco NAC and the TNC have become the two dominant competitors on the area. Juniper, Symantec and a large amount of smaller companies are committed to the TNC architecture while Cisco, as the leading networking equipment manufacturer, drives its own solution. However, it seems that Microsoft is trying to balance between these two parties. In September 2006 Microsoft and Cisco announced interoperability architecture for Cisco NAC and Microsoft NAP [50]. Less than a year later, in May 2007, Microsoft and TCG published a white paper describing interoperability between NAP and TNC [48] using the IF-TNCCS protocol bindings for Microsoft Statement of Health (SoH) protocol. Cisco has not agreed on supporting the TNC but it is driving the standardization of NAC-EI by contributing to IETF Network Endpoint Assessment Working Group (NEA-WG) and one of the two chairmen of the NEA work group is Susan Thompson from Cisco. However, the co-chair of the work group is Stephen Hanna from Juniper who is also the co-chair of the TNC Work Group. Because Juniper essentially represents the TNC, it looks like the NEA could be the point where these two competing architectures finally converge. As we discussed earlier, the NEA-WG is considering existing protocol implementations when defining the standards for NAC-EI. As of February 19 of 2008, which was the deadline for submitting proposals for NEA drafts, the TNC protocols IF-TNCCS and IF-M were the only proposals submitted. At the time of writing these drafts are at

version 6 which signifies that the work group is actively developing the drafts into RFCs , i.e. official Internet standards. Cisco has not submitted CNAC protocols as proposals for NEA standards which might be an indicator that TNC is going to win the standardization competition. It will be interesting to see whether it will enforce Cisco to adapt the TNC standards.

4.2 Case Studies

In this section we present a few case studies of implementing NAC solutions including posture-based authorization. St. Mary's County Public Schools in Maryland implemented a NAC solution to support a new program which involved 60 laptops connected into a wireless network. The security was a high concern so the school district implemented Juniper Unified Access Control (UAC) to enhance the security of the wireless network. IEEE 802.1X was chosen as the enforcement method and RADIUS as the authentication server. The agent is dynamically downloaded to the client computers and it provides capabilities for assessing the security state of the client including anti-malware, anti-virus and other security software. The result was that the school district was able to implement a highly secure wireless network by leveraging existing network infrastructure. [51]

The American University in Washington D.C. is a university with about 11000 students, 3000 faculty and approximately 6500 network devices. The university wanted to implement a NAC solution to mitigate the risk posed by misconfigured clients accessing the network. Cisco NAC Appliance was chosen as the NAC solution and a Cisco NAC client was deployed to each student computer accessing the network. As a result, the amount of malware tickets was reduced by 80 percent and the university was able to provide role-based access control and better methods to locate problematic clients. [52]

The Fulton County IT department is responsible for thousands of computers and supports approximately 5000 employees in 400 buildings. After suffering from network disruptions in the past due to non-compliant clients, the county decided to implement a new security solution. Microsoft NAP was chosen as the implementation. They first evaluated NAP using DHCP enforcement but decided that it didn't fulfill the security requirements. They ended up using IPsec enforcement to support also encryption of network communication. The county has moved from using cumbersome paper-based security policies to automated policy enforcement using NAP. According to the pilot implementation, the service-desk calls have decreased by 75 percent. [53]

Bangchak Petroleum, a Thai energy company, implemented NAC to conform to the ISO 27001 information security management standard. The company implemented various security mechanisms including Juniper UAC to support endpoint integrity inspection. The UAC enforces security policies for clients accessing both locally and remotely on the company network. [54]

St. Monica's college in Melbourne, Australia needed to build a secure intranet to support its learning management system as well as unified communication channels including voice, data and video. The college implemented the infrastructure using Juniper Networks' products including UAC for managing security policies. Using the NAC solution

the college was able to enforce extensive security policies using such criteria as groups, roles, individual users and computers, and time of day. [55]

4.3 Prospective

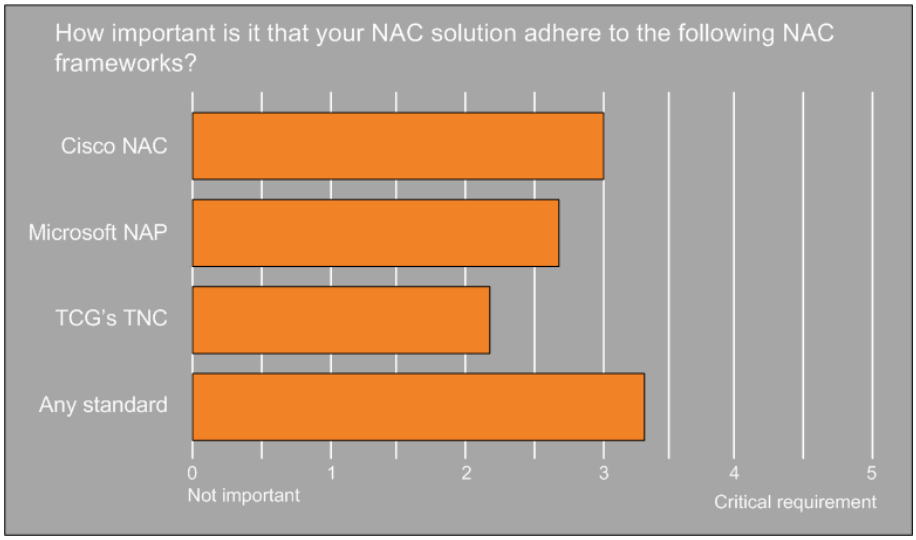

Figure 39: The importance of NAC solution adherence to an existing framework or standard [56]

According to a survey made by the Network Computing magazine in 2006, many organizations are evaluating different NAC implementations [56]. The survey comprised general NAC implementations, not only products offering posture-based authorization. The survey shows that in 2006 the NAC products that the organizations were most familiar with were Cisco NAC and Microsoft NAP. Only about 30 percent of the responders were evaluating the TNC whereas the same figure for CNAC and NAP was almost 40 percent. So the TNC does not have a very strong visibility among organizations. An interesting detail from the survey is that 48 percent of the responders indicated that they would not buy a NAC implementation until they understand Microsoft's role in the market. Microsoft's products are widely adopted in corporate IT infrastructures. By utilizing existing infrastructure when implementing integrity-based access control organizations may cut the costs of the implementation. If, for example, an organization's policy states that all client computers must have a Windows operating system installed, it means that by using Microsoft NAP as a NAC-EI solution the IT department does not need to deploy any additional client software to the client computer, as the NAP client is integrated into the newest Windows operating systems. NAP was officially released in 2008, two years after the survey, so the situation has probably changed a bit. The survey also examined how important it is for organizations that the implementation under evaluation adheres to a standard or an existing NAC framework. The results are summarized in Figure 39. The organizations preferred Cisco NAC over Microsoft NAP and TNC, but even more important was considered the adherence to any existing standards. This makes the competition

between Cisco and TNC interesting because now that it looks like that the TNC is becoming an IETF standard through NEA. According to the results presented in Figure 39 this might have a very positive impact on the adoption of TNC framework in organizations. [56]

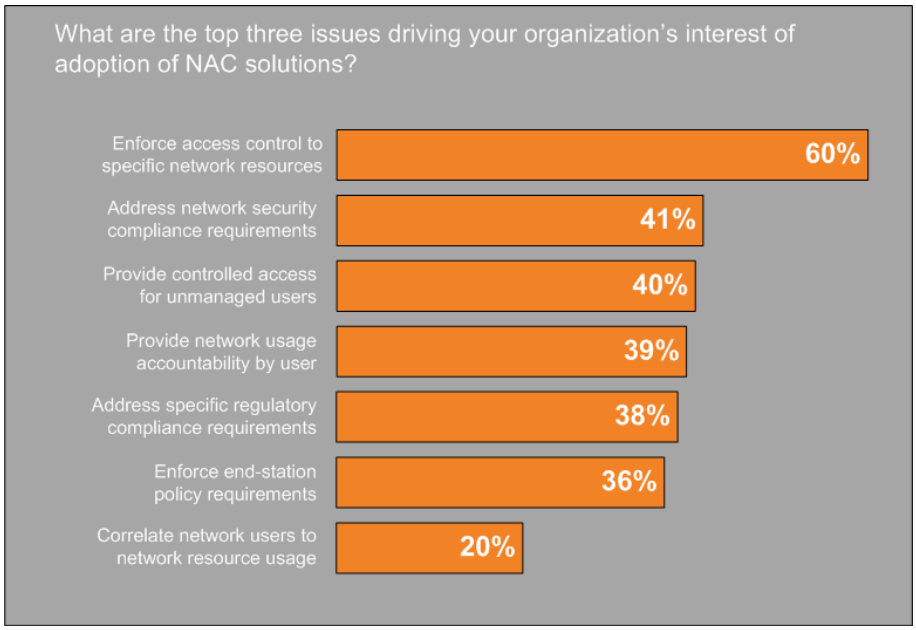

Figure 40: Issues that are driving organizations to adopt NAC solutions [56]

Now we have analyzed a little bit the racing between the different implementations of integrity-based access control. But is this a feature that is really needed and wanted in organizations? This is an important question to ask when analyzing the future of NAC-EI products because that is really what drives the market. During the last few years there has been a lot of discussion about new defense mechanisms for corporate networks. Recognized security evangelist Steve Riley held up a presentation called ”Death of the DMZ” a few years back. Riley's message during the last years has been that the attacks have moved from network and transport layers to application layer, and that is why the traditional way of protecting networks and endpoints using perimeter networks has become obsolete [57]. Because the employees are getting increasingly mobile it is getting increasingly important that the data can be accessed from everywhere. Because the clients are no longer only within the corporate network, the client cannot be trusted and the security must coexist in the servers with the data. But how would posture-based access control fit into this model? Well, the security should always be a made of layers. Using endpoint integrity as a factor for network access decisions is just providing additional layer of security. Figure 40 represents the issues that are driving organizations to provision NAC solutions according to [56]. According to the survey, by far the most important concern driving to implement NAC is the enforcement of access control policies to resources. The next most important issues are the ability to address compliance requirements and the ability to provide controlled access for unmanaged clients: contractors, partners and other guest users. These figures indicate that the organizations are willing to invest on NAC

products supporting integrity assessment. Providing access to contractors and partners is also a high priority issue which is actually the functionality provided by the federated TNC introduced earlier. However, the specification for the federated TNC is quite new and at the time of writing there are not any commercial implementations supporting it.

5 Conclusion

In this thesis we have examined architectures for supporting network access control based on endpoint integrity. We introduced Trusted Network Connect as an open standard architecture developed by Trusted Computing Group. TNC is designed to support existing network security standards, including RADIUS, EAP and IPsec. It is a feature-rich architecture supporting endpoint integrity verification, policy enforcement, monitoring, hardware-based authentication, and federation between security domains.

As a commercial implementation of integrity-based network access control we presented Microsoft Network Access Protection. NAP provides endpoint health assessment as well as isolation and remediation of non-compliant clients. It supports four different isolation methods, each of which is suitable for a bit different use case. Microsoft has also contributed to the TNC Work Group and in 2006 Microsoft and TCG announced interoperability between NAP and TNC.

When we compared the TNC and NAP architectures we found that the high-level architectures are almost identical with certain exceptions. TNC provides some features that are out of scope when it comes to NAP. These features include federated assessment between security domains, using Trusted Platform Module to further enhance security, and integration with a data store containing metadata information about clients in the network. The last few years have shown that the NAC solutions supporting integrity verification have divided into two camps: those that support TNC, and those that support Cisco Network Admission Control. Both sides have dozens of supporting vendors varying small and medium-sized into large enterprises such as Symantec and Juniper Networks. The main contributor of the TNC Work Group is Juniper Networks which is the main competitor of Cisco in the network equipment market. These two organizations are also driving the IETF Network Endpoint Assessment work group which aims at standardizing the NAC-EI architecture. As we saw, the NEA has recently adopted TNC protocols as drafts and is currently working on maturing these into RFCs, i.e. official Internet standards. We came into the conclusion that TNC is becoming the official standard in the industry even though the NEA contains only a subset of all the functionalities provided by TNC. This would be a big step towards interoperability and would probably increase the adoption of NAC-EI implementation. There are already a lot of case studies available of NAC-EI implementations, but it seems that many organizations have been postponing their decision of NAC implementation because of at least two things. The first and most important factor has been the lack of official industry standards. When there is no standardization there is always a risk of vendor lock-in, i.e. being dependent on products from only one vendor. Another important factor, as we saw from a survey made by Network Computing magazine, has been the role of Microsoft as a provider in this field. Almost every organization uses Microsoft products in their IT infrastructure. By utilizing existing infrastructure when implementing integrity-based access control organizations may cut the costs of the implementation. Of course this applies to other vendors also. If an organization has built its corporate network using switches and routers made by Cisco, it is probably a cost effective solution to use the same provider for NEC-EI also.

Even though it currently looks like the TNC is going to be the industry standard, it is

important to note that it does not automatically mean that all NAC-EI solutions will aim at compliance with the standard. Cisco has a strong alliance behind it and this would not be the first time when a large company like Cisco would use its market share to drive its own solution regardless of the industry standards. The next few years will show the direction of this competition.

The analysis that was made in this thesis about the future of NAC-EI was mostly based on case studies and a survey about NAC solution. The survey was made in 2006 so the information might not be accurate anymore but it was found useful when trying to find out the factors that affect the evaluation of different NAC solutions in organizations. In order to be able to better analyze the direction of the industry we would need a more accurate picture of the current situation in the organizations. This would require deeper study and more data gathered using e.g., a survey. Another interesting object for a study would be to compare the TNC and Cisco NAC architectures. This study might also help when analyzing the competition between these two frameworks. However, these studies are out of the scope of this thesis.

References

[1] Trusted Computing Group, Microsoft. Standardizing Network Access Control: TNC and Microsoft NAP to Interoperate, May 2007.

[2] Trusted Computing Group. Backgrounder, November 2006.

[3] TCG Trusted Network Connect. TNC Architecture for Interoperability, Specification Version 1.3, Revision 6, April 2008.

[4] Sangster, P., Khosravi, H., Mani, M., Narayan, K., Tardo J. IETF RFC 5209. Network Endpoint Assessment (NEA): Overview and Requirements, June 2008. Web document. Available at http://www.ietf.org/rfc/rfc5209.txt

[5] Sangster, P., Narayan, K. IETF Internet Draft. PA-TNC: A Posture Attribute Protocol (PA) Compatible with TNC, October 2009. Web document. Available at http://www.ietf.org/id/draft-ietf-nea-pa-tnc-06.txt

[6] Sahita, R., Hanna, S., Hurst, R., Narayan, K. IETF Internet Draft. PA-TNC: A Posture Attribute Protocol (PA) Compatible with TNC, October 2009. Web document. Available at http://www.ietf.org/id/draft-ietf-nea-pb-tnc-06.txt

[7] TCG Infrastructure Working Group Reference Architecture for Interoperability (Part I), Specification Version 1.0, Revision 1, June 2009.

[8] IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE Std 802.1X-2004, December 2004.

[9] TCG TNC Workgroup. TNC IF-IMC, Specification Version 1.2, Revision 8, February 2007.

[10] TCG TNC Workgroup. TNC IF-IMV, Specification Version 1.2, Revision 8, February 2007.

[11] TCG TNC Workgroup. TNC IF-TNCCS, Specification Version 1.1, Revision 1.00, February 2007.

[12] Alvestrand, H. IETF RFC 3066. Tags for the Identification of Languages, January 2001. Web document. Available at http://www.ietf.org/rfc/rfc3066.txt

[13] TCG TNC Workgroup. TNC IF-TNCCS: Protocol Bindings for SoH, Specification Version 1.0, Revision 0.08, May 2007.

[14] SMI Network Management Private Enterprise Codes. Web document. Available at http://www.iana.org/assignments/enterprise-numbers

[15] Adoba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H., Extensible Authentication Protocol (EAP). Internet Engineering Task Force, RFC 3748, June 2004. Web document. Available at http://www.ietf.org/rfc/rfc3748.txt

[16] Rigney, C., Willens, S., Rubens, A., Simpson, W., Remote Authentication Dial In User Service (RADIUS). Internet Engineering Task Force, RFC 2865, June 2000. Web document. Available at http://www.ietf.org/rfc/rfc2865.txt

[17] Simpson, W., The Point-to-Point Protocol (PPP). Internet Engineering Task Force, RFC 1661, July 1994. Web document. Available at http://www.ietf.org/rfc/rfc1661.txt

[18] Dierks, T., Rescorla, E., The Transport Layer Security (TLS) Protocol Version 1.2. Internet Engineering Task Force, RFC 5246, August 2008. Web document. Available at http://www.ietf.org/rfc/rfc5246.txt

[19] Simpson, W., PPP Challenge Handshake Authentication Protocol (CHAP). Internet Engineering Task Force, RFC 1994, August 1996. Web document. Available at http://www.ietf.org/rfc/rfc1994.txt

[20] Haller, N., Metz, C., A One-Time Password System. Internet Engineering Task Force, RFC 1938, May 1996. Web document. Available at http://www.ietf.org/rfc/rfc1938.txt

[21] Simon, D., Aboba, B., Hurst, R., The EAP-TLS Authentication Protocol. Internet Engineering Task Force, RFC 5216, March 2008. Web document. Available at http://www.ietf.org/rfc/rfc5216.txt

[22] TCG TNC Workgroup.TNC IF-M: TLV Binding, Specification Version 1.0, Revision 30, February 2008.

[23] TCG TNC Workgroup. TNC IF-T: Protocol Bindings for Tunneled EAP Methods, Specification Version 1.1, Revision 10, May 2007.

[24] Kaufman, C., Internet Key Exchange (IKEv2) Protocol. Internet Engineering Task Force, RFC 4306, December 2005. Web document. Available at http://www.ietf.org/rfc/rfc4306.txt

[25] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko, J., Diameter Base Protocol. Internet Engineering Task Force, RFC 3588, September 2003. Web document. Available at http://www.ietf.org/rfc/rfc3588.txt

[26] TCG TNC Workgroup. TNC IF-T: Binding to TLS, Specification Version 1.0, Revision 16, May 2009.

[27] ISO/IEC 11889. Trusted Platform Module (TPM).

[28] TCG Infrastructure Workgroup. Subject Key Attestation Evidence Extension, Specification Version 1.0, Revision 7, June 2005. Web document. Available at

[29] TCG TNC Workgroup. TNC IF-PEP: Protocol Bindings for RADIUS, Specification Version 1.1, Revision 0.7, February 2007.

[30] IEEE Standards for Local and Metropolitan Area Networks: Draft Standard for Virtual Bridged Local Area Networks, P802.1Q-2003, January 2003.

[31] IEEE Standards for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, IEEE Std 802.1D-2004, June 2004.

[32] Congdon, P., Sanchez, M., Aboba, B., RADIUS Attributes for Virtual LAN and Priority Support. Internet Engineering Task Force, RFC 4675, September 2006. Web document. Available at http://www.ietf.org/rfc/rfc4675.txt

[33] Congdon, P., Aboba, B., Smith, A., Zorn, G., Roese, J., IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines. Internet Engineering Task Force, RFC 3580, September 2003. Web document. Available at http://www.ietf.org/rfc/rfc3580.txt

[34] Leifer, D., Rubens, A., Shriver, J., Holdrege, M., Goyret, I., RADIUS Attributes for Tunnel Protocol Support. Internet Engineering Task Force, RFC 2868, June 2000. Web document. Available at http://www.ietf.org/rfc/rfc2868.txt

[35] Chiba, M., Dommety, G., Eklund, M., Mitton, D., Aboba, B., Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS). Internet Engineering Task Force, RFC 3576, July 2003. Web document. Available at http://www.ietf.org/rfc/rfc3576.txt

[36] TCG TNC Workgroup. TNC IF-MAP binding for SOAP, Specification Version 1.0, Revision 25, April 2008.

[37] W3C. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), April 2007.

[38] TCG WSS Workgroup. TCG Software Stack (TSS) Specification Version 1.2, Level 1, March 2007.

[39] TCG TNC Workgroup. Federated TNC, Specification Version 1.0, Revision 26, May 2009.

[40] Hughes, J., Maler, E., Lockhart, H., Wisniewki, T., Mishra, P., Ragouzis, N. Security Assertion Markup Language (SAML) 2.0 Technical Overview, May 2005.

[41] Microsoft Corporation. Network Access Protection Platform Architecture, February 2008.

[42] Microsoft Corporation. Internet Protocol Security Enforcement in the Network Access Protection Platform, February 2008.

[43] Palekar, A., Simon, Dan., Zorn, G., Josefsson, S., Protected EAP Protocol (PEAP). Internet Engineering Task Force, Internet Draft, March 2003. Web document. Available at http://tools.ietf.org/id/draft-josefsson-pppext-eap-tls-eap-06.txt

[44] Alexander, S., Droms, R., DHCP Options and BOOTP Vendor Extensions. Internet Engineering Task Force, RFC 2132, March 1997. Web document. Available at http://www.ietf.org/rfc/rfc2132.txt

[45] Droms, R., Dynamic Host Configuration Protocol. Internet Engineering Task Force, RFC 2131, March 1997. Web document. Available at http://www.ietf.org/rfc/rfc2131.txt

[46] Patel, B., Aboba, B., Dixon, W., Zorn, G., Booth, S., Securing L2TP using IPsec. Internet Engineering Task Force, RFC 3193, November 2001. Web document. Available at http://www.ietf.org/rfc/rfc3193.txt

[47] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., Zorn, G., Point-to-Point Tunneling Protocol (PPTP). Internet Engineering Task Force, RFC 2637, July 1999. Web document. Available at http://www.ietf.org/rfc/rfc2637.txt

[48] Microsoft Corporation, Trusted Computing Group. Standardizing Network Access Control: TNC and Microsoft NAP to Interoperate, July 2007.

[49] Cisco Systems. Cisco Network Admission Control (NAC) Executive Overview.

[50] Cisco Systems and Microsoft Corporation. Cisco Network Admission Control and Microsoft Network Access Protection Interoperability Architecture, September 2006.

[51] Juniper Networks. Case Study. St. Mary's County Publicic Schools Keep Students on the Forefront of Science and Technology with Digital Learning Secured by Unified Access Control, February 2009.

[52] Weakland, E. Network Access Control through Quarantine, Remediation, and Verification, May 2008. Web document. Available at http://www.educause.edu/Resources/NetworkAccessControlthroughQua/162942

[53] Microsoft Corporation. Case Study. Forward-Thinking County Government Enhances IT Security and Manageability, January 2008. Web document. Available at http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?CaseStudyID=4000001286

[54] Juniper Networks. Case Study. Juniniper Networks Puts Bangchak Petroleum on Track for ISO 27001 Certificication, February 2009.

[55] Juniper Networks. Case Study. St. Monica's College Builds State-of-the-art Network With Best-in-class Solutions From Juniper Networks, February 2009.

[56] Network Computing. NAC Vendors Square Off, July 2006.

[57] Steve Riley, Directly connect to your corpnet with IPsec and IPv6, June 2008. Web document. Available at http://blogs.technet.com/steriley/archive/2008/06/25/directly-connect-to-your-corpnet-with-ipsec-and-ipv6.aspx