[MS-SOH]:
Statement of Health for Network Access Protection (NAP) Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
10/22/2006	0.01		MCPP Milestone 1 Initial Availability
01/19/2007	1.0		MCPP Milestone 1
03/02/2007	1.1		Monthly release
04/03/2007	1.2		Monthly release

Date	Revision History	Revision Class	Comments
05/11/2007	1.3		Monthly release
06/01/2007	2.0	Major	Updated and revised the technical content.
07/03/2007	3.0	Major	Updated and revised the technical content.
07/20/2007	3.0.1	Editorial	Revised and edited the technical content.
08/10/2007	3.0.2	Editorial	Revised and edited the technical content.
09/28/2007	4.0	Major	Updated and revised the technical content.
10/23/2007	4.0.1	Editorial	Revised and edited the technical content.
11/30/2007	5.0	Major	Updated and revised the technical content.
01/25/2008	6.0	Major	Updated and revised the technical content.

Table of Contents

1 Introduction

This document specifies the Statement of Health for Network Access Protection (NAP) Protocol in which a client and a server exchange **Statement of Health (SoH)** and **Statement of Health Response (SoHR)** messages. This protocol, along with appropriate **authentication** protocols, helps enterprises to ensure that users of their network resources are not only authenticated but are using systems that conform with corporate **policies**. Typically the policies relevant to this protocol relate to security update management, configuration for antivirus products, firewall settings, and similar security/system health measures.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Authentication
Authorization
Computer Name
EAP Server
FILETIME
Fix-Up Servers
Health ID
Health Messages
Health Policy Server
Health Registration Authority (HRA)
Health State
HRESULT
Mandatory TLV
Network Access Server (NAS)
Policy
Remediation Server
Remote Authentication Dial-In User Service (RADIUS)
Session
SoH Client
Statement of Health (SoH)
Statement of Health Response (SoHR)
System Health Entity
Uniform Resource Locator (URL)

The following terms are specific to this document:

Health Certificate Enrollment Protocol (HCEP): A protocol designed to accomplish health certificate enrollment. Health certificates encapsulate the client's compliance to **policy** in a way that can be presented to interested parties without requiring those parties to perform the validation themselves.

IANA SMI: Structure and Identification of Management Information for TCP/IP-based Internets (SMI), a data structure defined by Internet Assigned Numbers Authority (IANA) to manage hosts and gateways on the Internet. As specified in [IANA-ENT], [IANA-NMP], and [RFC1155].

Idempotence: An operation where, if the operation is applied one or more times, no difference, errors, or inconsistencies will result. Example: abs(x) == abs(abs(x)) == abs(abs(abs(x))) == ... for all x.

Man-In-The-Middle (MITM) Attack: A security attack in which an attacker intercepts and possibly modifies data that is transmitted between two users. The attacker pretends to be the

other person to each user. In a successful **man-in-the-middle attack**, the users are unaware that there is an attacker between them, intercepting and modifying their data. Also referred to as a bucket brigade attack.

Statement of Health ReportEntry (SoH ReportEntry): A collection of data that represents a specific aspect of the **health state** of a client.

Statement of Health Response ReportEntry (SoHR ReportEntry): A collection of data that represents the evaluation of a specific aspect of the **health state** of a client, according to network **policies**.

System Health Agent (SHA): The client components that make declarations on a specific aspect of the client **health state** and generate an **SoH ReportEntry**.

System Health Validator (SHV): The server counterpart to the **System Health Agent (SHA)**, which is responsible for verifying the declarations of client **health state** made by the respective **SHA**. The **SHV** generates an **SoHR ReportEntry**.

Type-Length-Value (TLV): An information element that is encoded within a protocol. Type and Length fields are a fixed size (1 to 4 bytes), and the Value field is variable. Type indicates what kind of field is encoded; Length indicates the size of Value; and Value defines the data portion of this **Type-Length-Value (TLV)** element.

Type-Value (TV): An information element that is encoded within a protocol. The Type field is of a fixed size. Type indicates both what kind of value is encoded and the length of the Value field (by implication). This is because each type in a **Type-Value (TV)** is of a fixed and known length.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[IANA-ENT] Internet Assigned Numbers Authority, "Private Enterprise Numbers", January 2007, http://www.iana.org/assignments/enterprise-numbers

[IANA-NMP] Internet Assigned Numbers Authority, "Network Management Parameters", http://www.iana.org/assignments/smi-numbers

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-HCEP] Microsoft Corporation, "Health Certificate Enrollment Protocol Specification", January 2007.

[MS-RNAP] Microsoft Corporation, "Vendor-Specific RADIUS Attributes for Network Access Protection (NAP) Data Structure", January 2007.

[MS-WSH] Microsoft Corporation, "Windows Security Health Agent (WSHA) and Windows Security Health Validator (WSHV) Protocol Specification", July 2007.

[RFC1155] Rose, M. and McCloghrie, K., "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, May 1990, http://www.ietf.org/rfc/rfc1155.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2781] Hoffman, P. and Yergeau, F., "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000, http://www.ietf.org/rfc/rfc2781.txt

[RFC2865] Rigney, C., Willens, S., Rubens, A., and Simpson, W., "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000, http://www.ietf.org/rfc/rfc2865.txt

1.2.2 Informative References

[MS-PEAP] Microsoft Corporation, "Protected Extensible Authentication Protocol (PEAP) Specification", January 2007.

[MSDN-OSVERSIONINFOEX] Microsoft Corporation, "OSVERSIONINFOEX Structure", http://msdn2.microsoft.com/en-us/library/ms724833.aspx

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997, http://www.ietf.org/rfc/rfc2131.txt

[RFC2409] Harkins, D. and Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998, http://www.ietf.org/rfc/rfc2409.txt

1.3 Protocol Overview (Synopsis)

It is common for network administrators to require authentication and **authorization** for users or devices attaching to their networks. Likewise, administrators require that the devices conform with the security policies of the organization if they are to access the network. For example, an administrator might require that every client accessing the network have a host firewall configured in a particular manner to protect both the network and the client computer.

There are a number of ways in which an administrator can approach this particular problem. Historically, the most common way has been through the creation of written policies that administrators hope users will follow.

This approach only works well with small groups of trustworthy and technically adept personnel. It fails in scenarios where large groups of users are involved. Many users are either unable or unwilling to follow the prescribed policies.

Network Access Protection (NAP) is a system developed by Microsoft to provide a more reliable alternative to this problem. NAP uses the Statement of Health for NAP Protocol to manage a computer's conformance with corporate security policies. The Statement of Health for NAP Protocol uses Statement of Health (SoH) and Statement of Health Response (SoHR) messages exchanged between a client and a server to validate client conformance with corporate security policies. This protocol can be used in any other mechanism intended to manage the health of connected resources.

Note The term Statement of Health (SoH) occurs both in the name of the protocol and the name of the message from the client to the server. In the interest of clarity, whenever Statement of Health is used to refer to the protocol, the phrase Statement of Health for NAP Protocol will be used.

The notion of health in this context has to do with conformance to corporate policies for how a system should be configured. A system is healthy if it conforms to corporate policy. It is not healthy if it does not conform to policy.

The purpose of the SoH message is to report the **health state** of the client that is in the process of requesting access to a network resource. SoH messages are typically exchanged as part of the process to authenticate and authorize the client or user. The client uses the SoH to report its state so it can be evaluated against the corporate policy.

The purpose of the Statement of Health Response (SoHR) is two-fold. It indicates whether the client meets policy requirements based on evaluation of the SoH. This allows the server/service to allow/disallow the connection request. Additionally, the SoHR communicates to the client what, if any, measures it must take in order to conform with policy, in the event that it is not conformant.

These messages may be carried within other authentication and authorization protocols, such as the **Health Certificate Enrollment Protocol (HCEP)**, as specified in [MS-HCEP] section 1.2. Carrying the Statement of Health for NAP Protocol in a higher-level transport protocol that has built-in security measures has the advantage of securing the Statement of Health for NAP Protocol.

The Statement of Health for NAP Protocol is a simple protocol in which there is a single exchange between the client and the server. The flow is as follows in the case of its successful use in HCEP:

1. The client sends an SoH inside an HCEP message that is posted to a **health registration authority (HRA)**.
2. The SoH is then checked for conformance with network policies.
3. The result is encoded in an SoHR.
4. The HRA replies to the client with an HCEP message that includes the SoHR inside.

This process is as specified in [MS-HCEP] section 1.3.

Note The processing done in steps 2 and 3 are specific to the protocol that carries the SoH/SoHR messages.

1.4 Relationship to Other Protocols

The Statement of Health for NAP Protocol can run on any protocol; however, it is most commonly used in protocols that perform network-level authentication such as the Protected Extensible Authentication Protocol, as specified in [MS-PEAP].

The protocol also uses services and applications (**system health agents (SHA)**) on the client side to produce the information that must be included in the SoH. Similarly, on the server side, the protocol uses services (**system health validators (SHV)**) that process and evaluate the SoH and produce an SoHR in response. The interface between the component implementing the protocol and the component providing these services is an implementation choice. One specific SHA and SHV are included in Windows to report and evaluate system health state respectively. These are the Windows Security Health Agent and Windows Security Health Validator (WSHA/WSHV), as specified in [MS-WSH]. Notice that SHAs and SHVs can be implemented by third-parties.

1.5 Prerequisites/Preconditions

For a Statement of Health for NAP Protocol exchange to occur, the **SoH Client** must have a **session** with a suitable transport protocol established with a **Health Policy Server** or to an intermediary server that can relay the Statement of Health for NAP Protocol to the Health Policy Server.

As a precondition, the SoH Client is required to be able to construct valid SoH messages in the format defined in section 2. It is also a requirement that the client be able to process well-formed SoHR messages. Similarly, as a precondition, the SoH server must be able to process SoH messages and construct SoHR messages.

The actual content of the SoH messages is implementation specific.

As explained in section 1.7, there are two protocol versions. The SoH Client should send version 2 messages. The SoH server must accept version 2 messages and may accept version 1 messages. The SoH server must create a response that matches the version of the received request.<1>

The most common use for the Statement of Health for NAP Protocol is one in which a client connects to a **network access server (NAS)** and the NAS connects as a **Remote Authentication Dial-In User Service (RADIUS)** client to a RADIUS server. This scenario is specified in [MS-RNAP] section 1.3. Each vendor specifies how the transport handles the SoH.<2>

1.6 Applicability Statement

The Statement of Health for NAP Protocol is designed to provide to enterprises a mechanism that helps ensure the systems they manage are healthy.

The protocol may be used in conjunction with an authentication protocol for network access. It is also possible to use the protocol independently of any authentication process. Thus, the protocol does not need to be tied to authentication and authorization but may simply be used to manage the health of the enterprise computing resources. An example of such usage is when DHCP is used as the carrier of SoH or SoHR messages.

Another way of using the Statement of Health for NAP Protocol is to obtain a credential (for example, a X.509 certificate) using an enrollment process that includes this protocol. Possessing such a credential is the equivalent of having had the client evaluated to be in good health. The Health Certificate Enrollment Protocol (HCEP), as specified in [MS-HCEP], is an example of this usage.

The result of the Health Certificate Enrollment Protocol, in the successful case, is to provision the client with a X.509 certificate that can be used in an authentication protocol such as the Internet Key Exchange (IKE) Protocol (for more information, see [RFC2409]).

1.7 Versioning and Capability Negotiation

The Statement of Health for NAP Protocol has two versions. A version 2 message differs from a version 1 message in that it has an SoH mode sub-header, as specified in section 2.2.7. This SoH mode subheader is intended to allow a later version of the protocol to have newer modes of operation. There is no other functional difference between the two.

When a server receives a request message from a client, it creates a response with the same version as the request. The version of the message affects only the headers, as defined in sections 2.2.5 and 2.2.6.

1.8 Vendor-Extensible Fields

The Statement of Health for NAP Protocol provides a single vendor-extensible field in its SoH and SoHR messages (see section 2.2.3.3).

1.9 Standards Assignments

Parameter	Value	Reference
IANA SMI Vendor ID for Microsoft	0x137 (Decimal 311)	[IANA-ENT]

2 Messages

The following sections specify transport and the syntax of attributes for the Statement of Health for NAP Protocol.

2.1 Transport

The Statement of Health for NAP Protocol does not provide its own transport. It MUST be carried in some other protocol that provides transport for it. For instance, it may be carried in HCEP, as specified in [MS-HCEP], or in RADIUS, as specified in [MS-RNAP] section 2.2.1.8.

2.2 Message Syntax

The SoH and the SoHR are identical structures. They are composed of a header followed by a set of **Type-Length-Values (TLVs)**.

The first TLV in every SoH/SoHR is the System-Health-ID TLV of the System Statement of Health (SSoH)/System Statement of Health Response (SSoHR). The value part of the SSoH/SSoHR is a sequence of **Type-Value (TV)** structures. Allowable values of the TV structures are defined later in this section.

The TLVs that follow the SSoH/SSoHR are grouped in SoHReportEntry/SoHRReportEntry sets. The System-Health-ID TLV marks the beginning of each set of TLVs that constitute an SoHReportEntry/SoHRReportEntry, and MUST be present. Each TLV in an SoHReportEntry/SoHRReportEntry is called an SoHAttribute/SoHRAttribute. There MAY be zero or more SoHAttributes per SoHReportEntry (see section 2.2.5.2) besides the System-Health-ID TLV. In addition to the System-Health-ID TLV (see section 2.2.6.2), there MUST be one or more SoHRAttributes per SoHRReportEntry.

These structures and their allowable values are defined below. The fields of these structures are transmitted in network-byte order from left to right as shown in the following figures and tables throughout this section.

A graphic representation of the top-level structure follows.

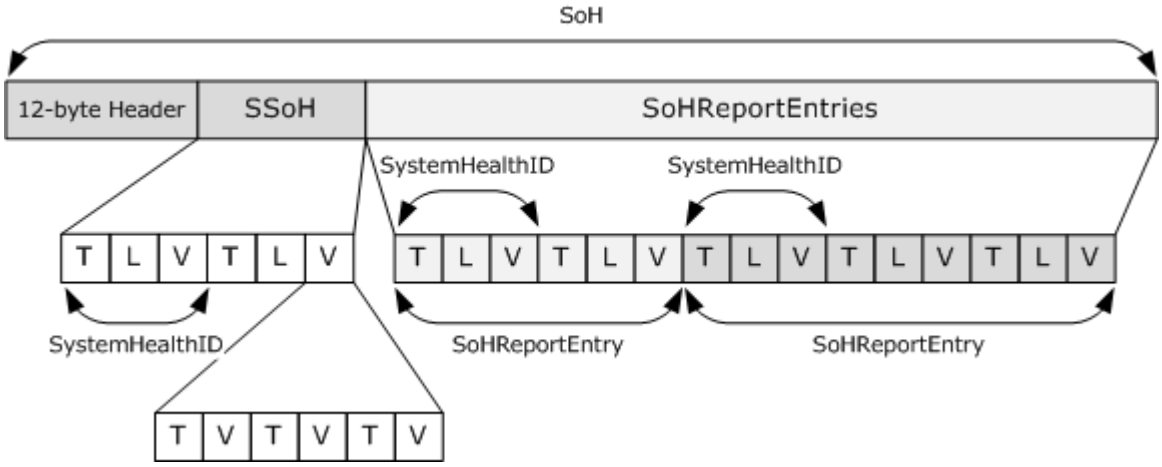


Figure 1: SoH shown without sub-mode header

See section 2.2.7.

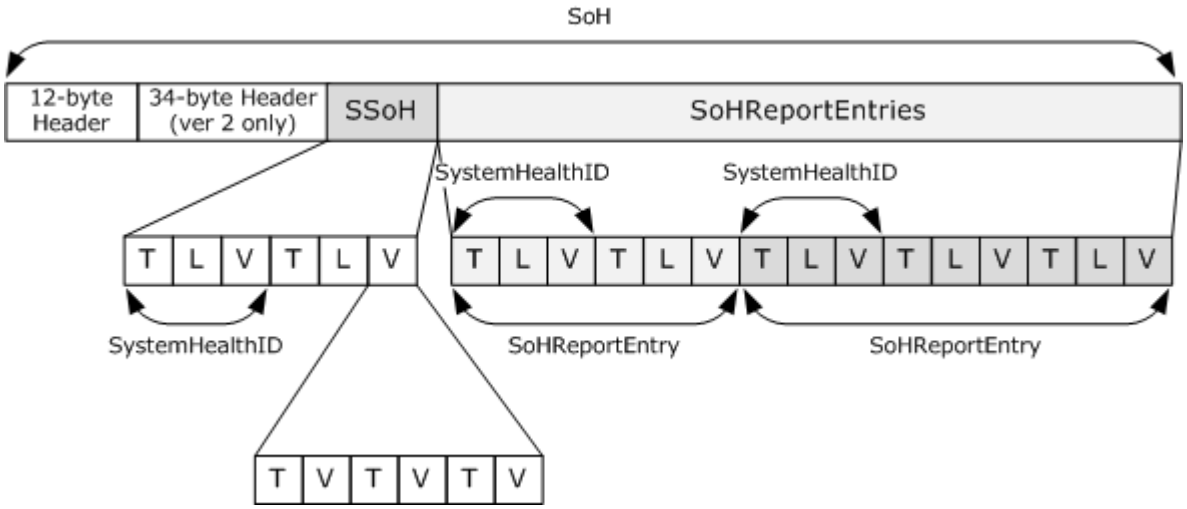


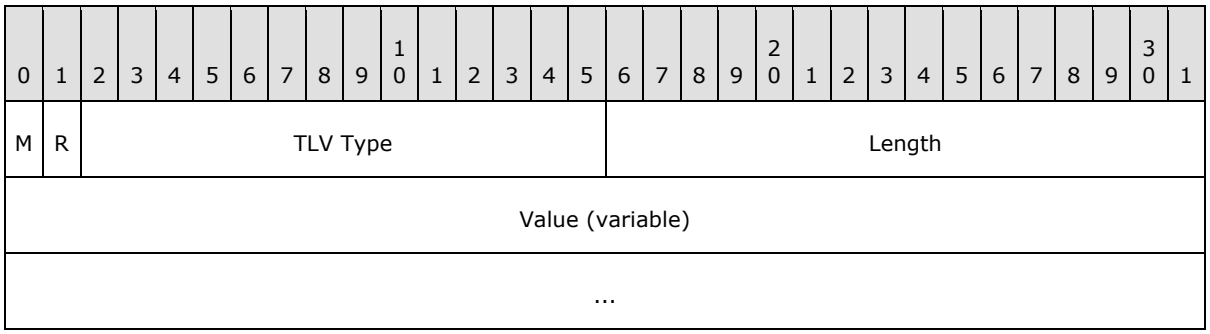
Figure 2: SoH shown with sub-mode header

See section 2.2.7.

2.2.1 Type-Length-Value (TLV) Packet

The following diagram specifies the TLV (**M**, **R**, **TLV Type**, **Length**, and **Value**) structure that forms the basis for SoH/SoHR messages.

The Type-Length-Value Packet



M (1 bit): The **M** bit has the following possible values, and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a **mandatory TLV**.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

TLV Type (14 bits): A 14-bit unsigned integer that MUST indicate the type of data in the **Value** field. Valid **TLV Type** values MUST be one of those specified in section 2.2.3.

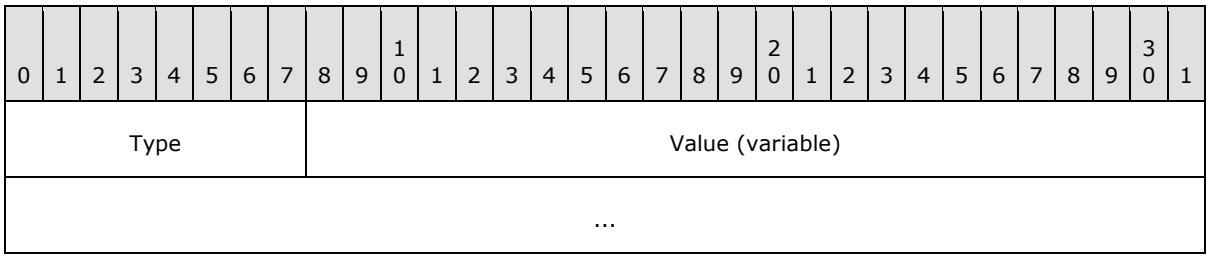
Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): The value MUST be formatted in accordance with the type specified in the **TLV Type** field.

2.2.2 Type-Value (TV) Packet

The following diagram shows the standard Type-Value (TV) structure that is used by all System Statement of Health (SSoH)/System Statement of Health Response (SSoHR) attributes, as specified in section 2.2.4.

The Type-Value Packet



Type (1 byte): An 8-bit unsigned integer that indicates the type of data in the attribute **Value** field.

Value (variable): The length of the **Value** MUST correspond to the type as defined by the **Type**. Each type is of a fixed length, although different types have different values.

2.2.3 SoHAttributes / SoHRAttributes

The SoHAttribute/SoHRAttribute elements are messages that are used to construct valid SoH/SoHR messages. A collection of SoHAttributes/SoHRAttributes in which the first attribute is the System-Health-ID constitutes an SoHReportEntry/SoHRReportEntry.

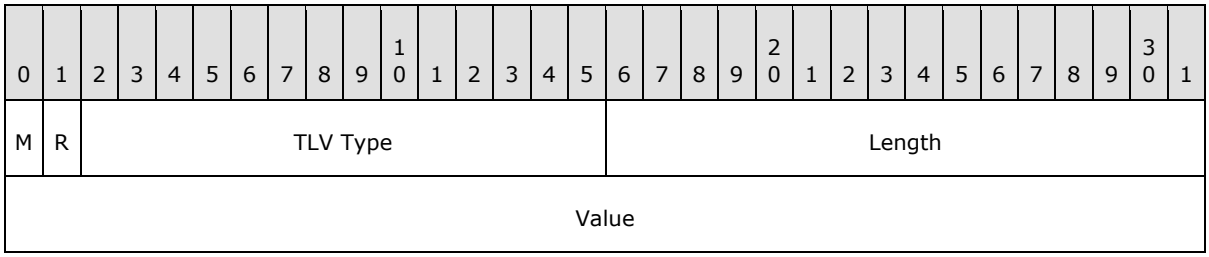
When using a TLV to contain an SoH/SoHR attribute, the TLV types, with values between and including 0-256, are reserved for use in the Statement of Health for NAP Protocol, and MUST NOT be used for other attributes. The following TLV types MUST be supported for use within this protocol:

Type	Name	Meaning
2	System-Health-ID	ID of the system health agent (SHA) or system health validator (SHV) that generated the SoHReportEntry or SoHRReportEntry.
4	Compliance-Result-Codes	Result codes specifying whether or not the client computer is compliant.
7	Vendor-Specific	The **Value** field contains a vendor-specific TLV.
14	Failure Category	A code that indicates the Failure Category.

A list of optional TLVs are provided in section 2.2.3.5.

2.2.3.1 System-Health-ID Packet

The ID of the component that generated the SoHReportEntry/SoHRReportEntry (for the SoH or SoHR) MUST be the first TLV present in the SoHReportEntry/SoHRReportEntry.



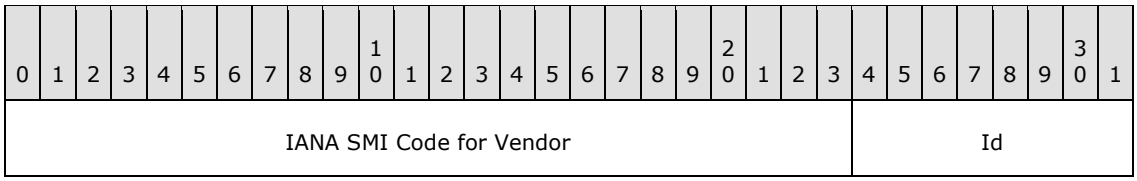
M (1 bit): The **M** bit MUST be set to zero.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

TLV Type (14 bits): The **TLV Type** for this packet type MUST be set to 2.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field. For this packet type, MUST be set to 4.

Value (4 bytes): A 32-bit unsigned integer used to represent the **Health ID** of the SoHReportEntry/SoHRReportEntry. The value MUST be formatted as follows:



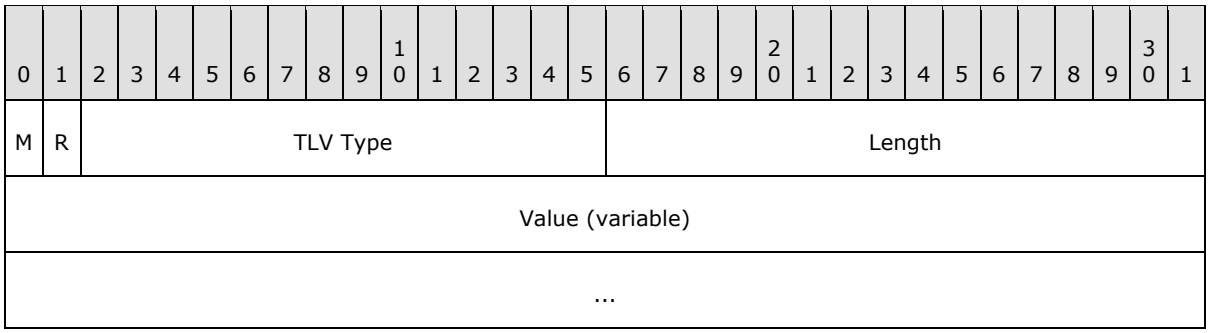
IANA SMI Code for Vendor (3 bytes): A 24-bit unsigned integer that MUST contain the IANA SMI code for the vendor whose component produced the message.

Id (1 byte): An 8-bit unsigned integer used to identify different components from the same vendor. Any value can be specified by the vendor for use by its components.<3>

2.2.3.2 Compliance-Result-Codes

The result of the evaluation of the SoH is used to specify whether the client machine is compliant with policy. An SoHR MUST contain this attribute, a Failure Category attribute, or both. An SoH MAY contain this attribute.

The TLV values of a Compliance-Result-Codes attribute are as follows:



M (1 bit): The **M** bit MUST be set to zero.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

TLV Type (14 bits): The **TLV Type** for this packet type MUST be set to 4.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): An array of **HRESULTs** that indicate the results of evaluation by the server.

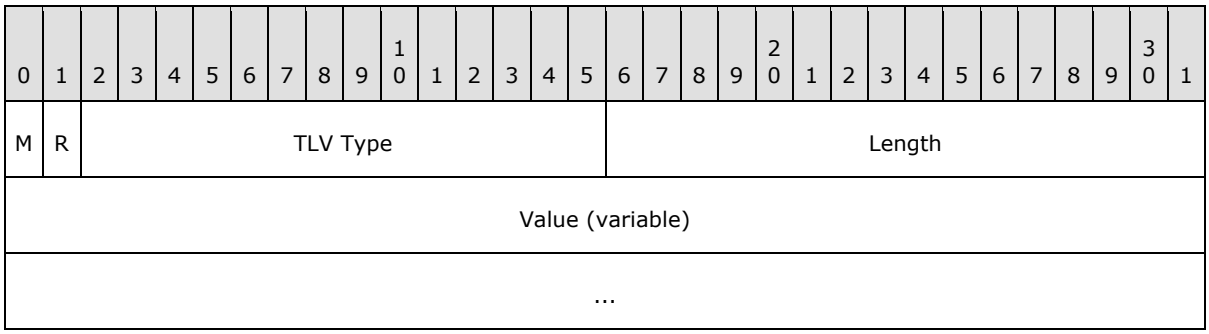
2.2.3.3 Vendor-Specific Packet

The Vendor-Specific Packet represents vendor-specific data, in which the format of the data is known only to the vendor.

This attribute is used to carry implementation-specific data from the client to the server and back. For example, an antivirus vendor may include data about the signature database version and the time at which the last complete scan was performed.

This attribute can be present in an SoH and SoHR.

The TLV values of a Vendor-Specific attribute are as follows:



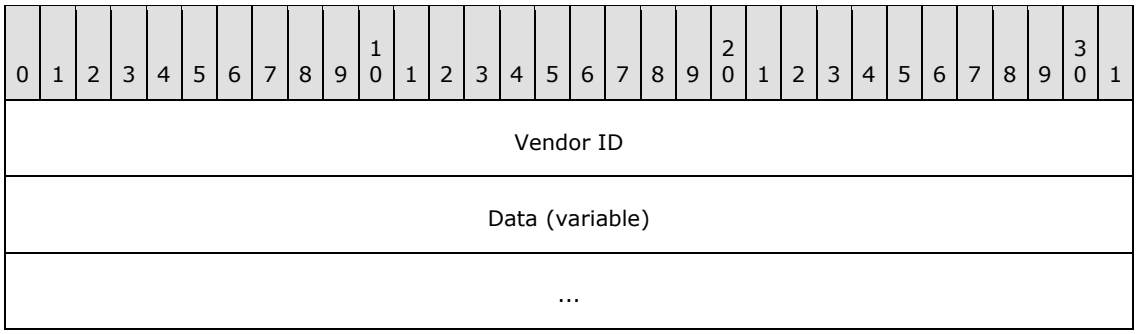
M (1 bit): The **M** bit MUST be set to zero.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 7.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): The **Value** field MUST be formatted as follows:



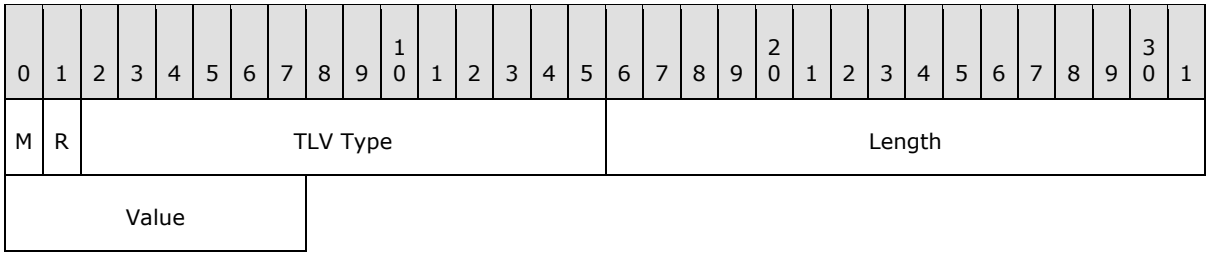
Vendor ID (4 bytes): 32-bit unsigned integer that SHOULD specify the IANA-assigned SMI for the vendor whose data is to be specified in the **Data** field.<4> NAP does not interpret this field. The vendor MAY use it for any purpose.

Data (variable): The format of the **Data** field is vendor specific. An example can be found in the MSSHA implementation in WSHA SoH and WSHV SoHR.

2.2.3.4 Failure Category

The Failure Category attribute is used to classify the type of failure that occurred. An SoHR MUST contain this TLV, a Compliance-Result-Code TLV, or both. This attribute MAY be present in an SoH.

The TLV values of a Failure Category attribute are as follows:



M (1 bit): The **M** bit MUST be set to zero.

R (1 bit): The **R** bit is reserved and MUST be set to zero and ignored on receipt.

TLV Type (14 bits): The **TLV Type** MUST be set to 14.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field. For this packet type, MUST be set to 1.

Value (1 byte): An 8-bit field that MUST contain one of the following values:

Value	Meaning
0	No failure occurred.
1	Failure that is not due to client or server components or communications.
2	Failure due to client component.

Value	Meaning
3	Failure due to client communication.
4	Failure due to server component.
5	Failure due to server communication.

2.2.3.5 Optional TLVs

The following table contains a list of optional TLVs that can be used by an implementation. The TLVs that are not optional are excluded from the table below and can be found in section 2.2.3. These TLVs can be present in an SoH or SoHR message. If these types are used to construct SoHs/SoHRs, the lengths specified in the following sections for each optional TLV MUST be honored.<5>

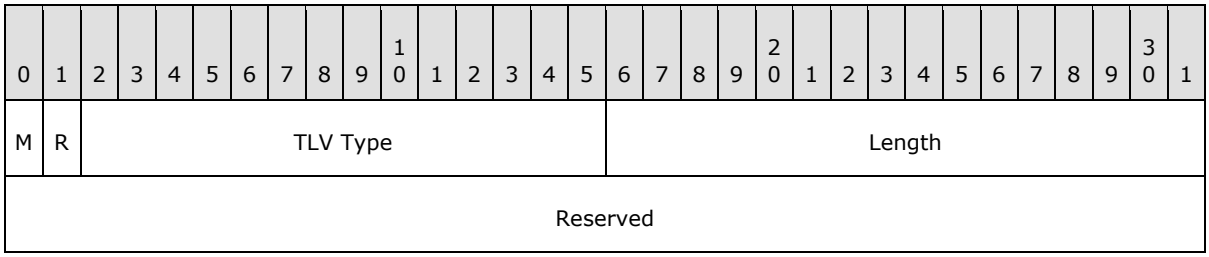
Type	Name	Value	Length in bytes
0	Reserved, specified in section 2.2.3.5.1.	Reserved	4
1	Reserved, specified in section 2.2.3.5.2.	Reserved	4
3	IPv4-Fixup-Servers, specified in section 2.2.3.5.3.	IPV4 addresses of the **fix-up servers**	Variable
5	Time-of-Last-Update, specified in section 2.2.3.5.4.	UTC time when client machine was last updated (measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC))	8
6	Client-Id, specified in section 2.2.3.5.5.	Identifier for the client	Variable
8	Health-Class, specified in section 2.2.3.5.6.	Type of health check that the SHA is performing (firewall, antivirus, critical update, and so on)	1
9	Software-Version, specified in section 2.2.3.5.7.	Version of the software installed on the client computer	1
10	Product-Name, specified in section 2.2.3.5.8.	Name of the product installed on the client computer	Variable
11	Health Class Status, specified in section 2.2.3.5.9.	Status code for the health-class type given by the Health-Class TLV	Variable
12	SoHGenerationTime, specified in section 2.2.3.5.10.	UTC time when the SoH was generated	8
13	Error Codes, specified in section 2.2.3.5.11.	Error codes for specific operations that can be contained in the SoHReportEntry or the SoHRReportEntry	Variable
15	IPv6-Fixup Servers, specified in section 2.2.3.5.12.	IPV6 addresses of the fix-up servers	Variable
16-	Reserved	Reserved	N/A

Type	Name	Value	Length in bytes
255			

The following sections detail the defined optional TLVs 0-15.

2.2.3.5.1 Optional TLV 0: Reserved

The 0 Reserved packet is reserved for future use, and MUST be ignored if received in error.



M (1 bit): The **M** bit has the following possible values, and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

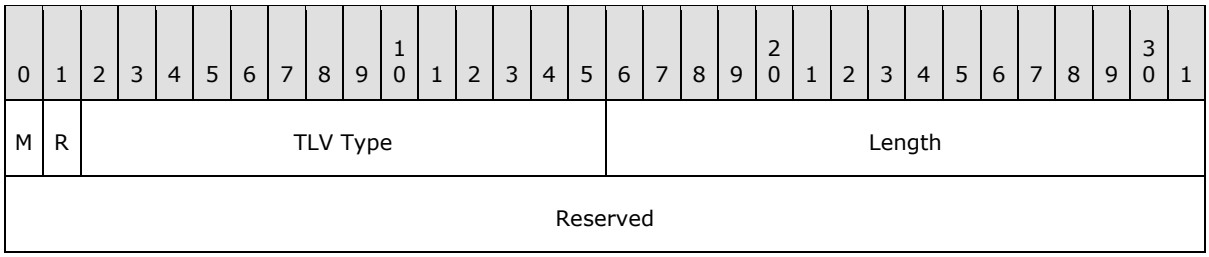
TLV Type (14 bits): Reserved. MUST always be 0.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the Reserved field. For this packet type, MUST be set to 4.

Reserved (4 bytes): Reserved for future use. MUST be 0, and MUST be ignored if received in error.

2.2.3.5.2 Optional TLV 1: Reserved

The 1 Reserved packet is reserved for future use and MUST be ignored if received in error.



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved and MUST be set to zero and ignored on receipt.

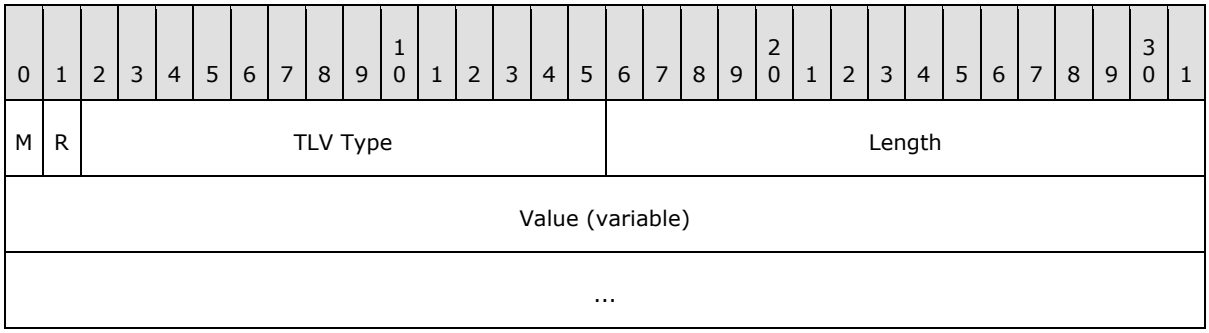
TLV Type (14 bits): Reserved. For this packet type, MUST always be 1.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the Reserved field. For this packet type, MUST be set to 4.

Reserved (4 bytes): Reserved for future use. MUST be 0 and MUST be ignored if received in error.

2.2.3.5.3 Optional TLV 3: IPv4-Fixup-Servers

The IPv4-Fixup-Servers packet provides the addresses of the fix-up servers. An SoH MAY contain 0+ this attribute. An SoHR MAY contain 0 or 1 this attribute. A SHA may use this information to perform remediation.



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

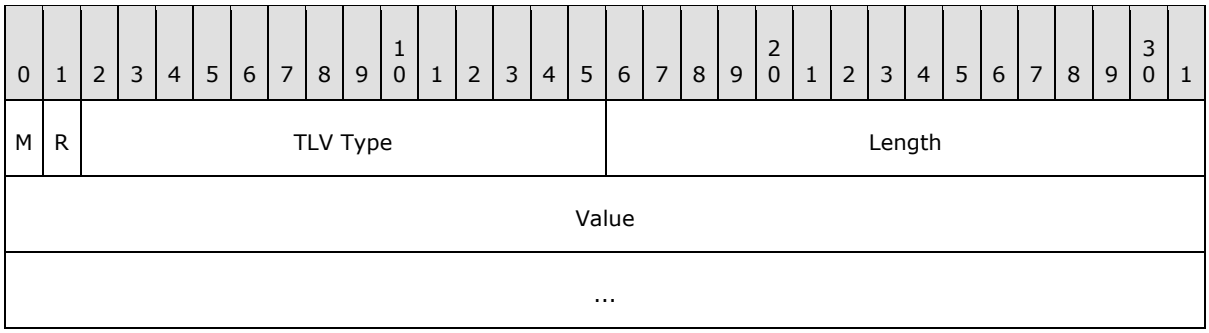
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 3.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): An array containing the four-byte IPV4 addresses of the fix-up servers.

2.2.3.5.4 Optional TLV 5: Time-of-Last-Update

The Time-of-Last-Update packet specifies the UTC time when the client machine was last updated (measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC)).<6>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

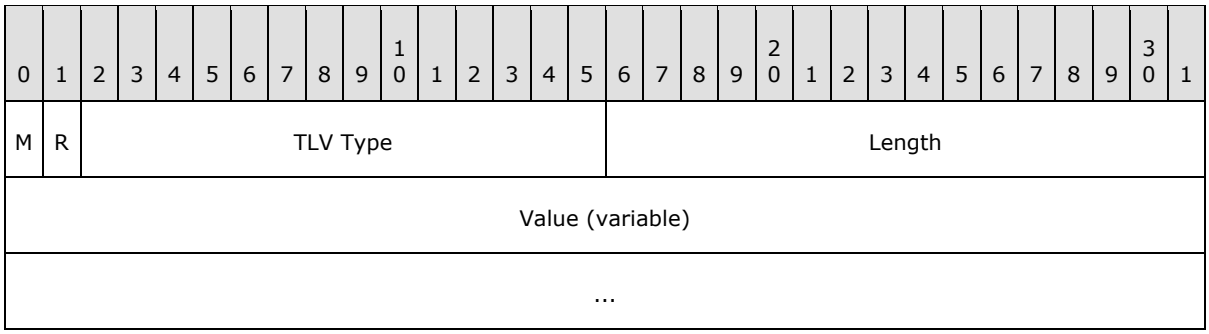
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 5.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field. For this packet type, MUST be set to 8.

Value (8 bytes): MUST be the UTC time when the client machine was last updated (measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC)).

2.2.3.5.5 Optional TLV 6: Client-ID

The Client-ID packet specifies the client identifier.<7>



M (1 bit): The **M** bit has the following possible values, and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

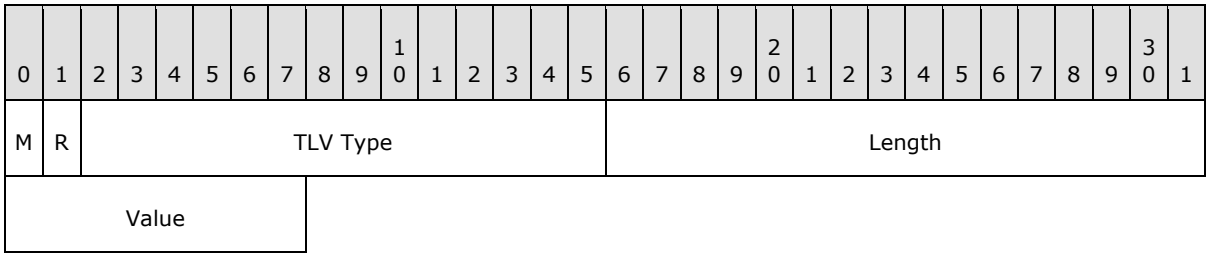
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 6.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): MUST specify the client identifier as a null-terminated string.

2.2.3.5.6 Optional TLV 8: Health-Class

The Health-Class packet specifies the type of health check that the SHA is performing (firewall, antivirus, critical update, and so on).<8>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

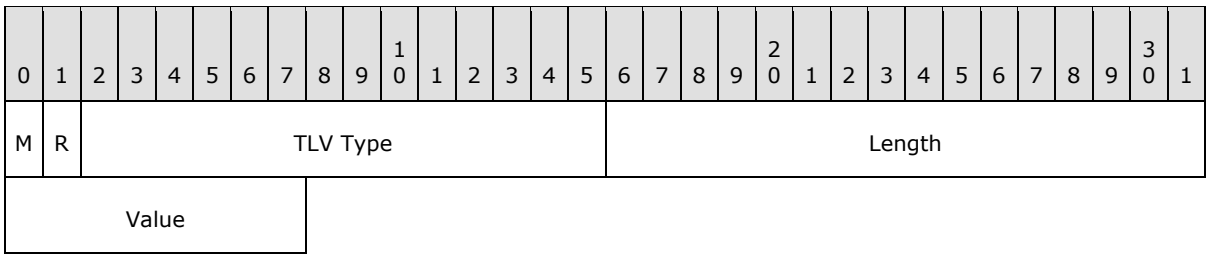
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 8.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field. For this packet type, MUST be set to 1.

Value (1 byte): An 8-bit field that MUST specify the type of health check that the SHA is performing (firewall, antivirus, critical update, and so on). An example can be found in the MSSHA implementation in WSHA SoH and WSHV SoHR.

2.2.3.5.7 Optional TLV 9: Software-Version

The Software-Version packet specifies the version of the software installed on the client computer.<9>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

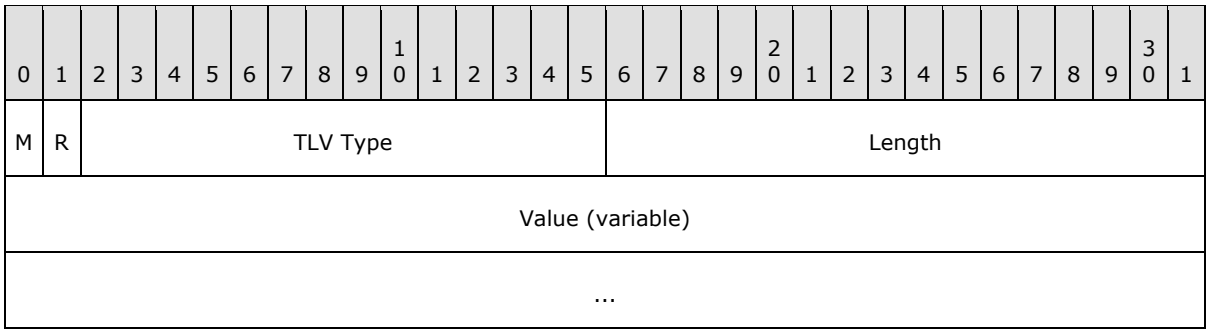
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 9.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field. For this packet type, MUST be set to 1.

Value (1 byte): An 8-bit field specifying the version of the software installed on the client computer.

2.2.3.5.8 Optional TLV 10: Product-Name

The Product-Name packet specifies the name of the product installed on the client computer.<10>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

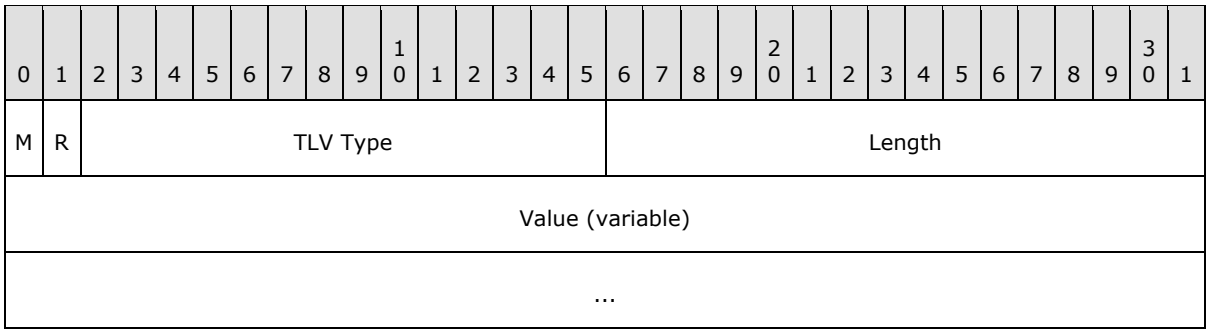
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 10.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): A null-terminated string that MUST specify the name of the product installed on the client computer.

2.2.3.5.9 Optional TLV 11: Health Class Status

The Health Class Status packet specifies the Status code for the health-class type given by the Health-Class TLV.<11>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

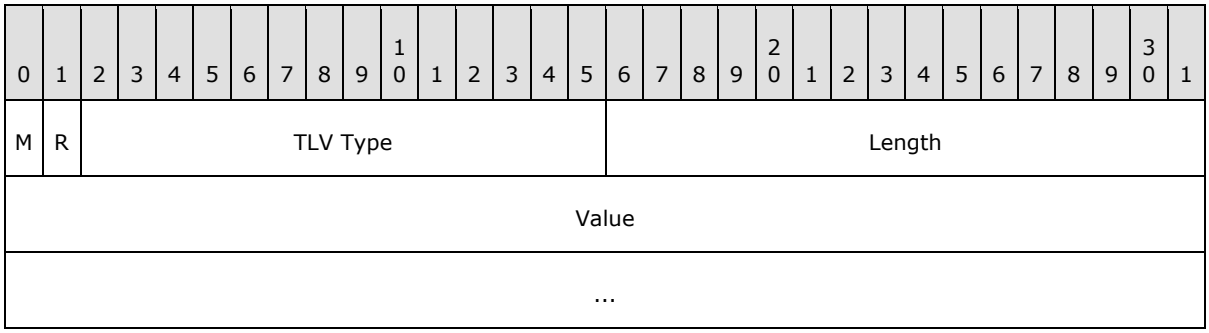
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 11.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): A null-terminated string (a sequence of bytes) that MUST specify the status code for the health-class type given by the Health-Class TLV. An example can be found in the MSSHA implementation in WSHA SoH and WSHV SoHR.

2.2.3.5.10 Optional TLV 12: SOH Generation Time

The SOH Generation Time packet specifies the UTC time when the SoH was generated.<12>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

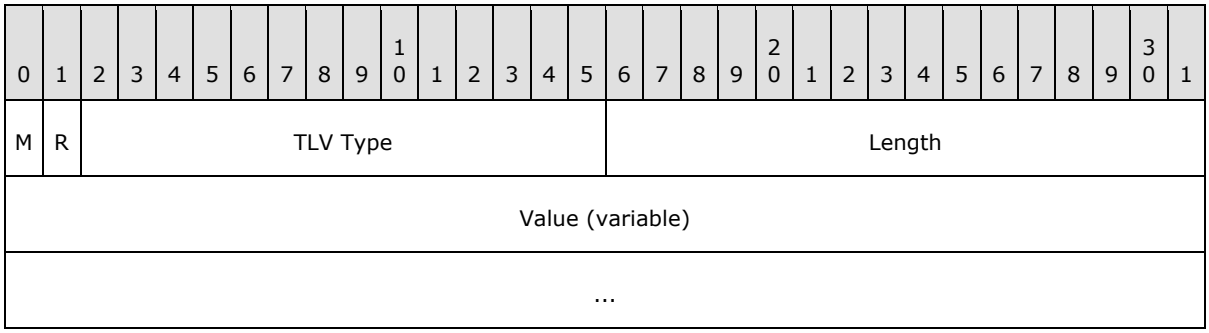
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 12.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field. For this packet type, MUST be set to 8.

Value (8 bytes): Specifies the UTC time when the SoH was generated. This 64-bit value MUST represent the number of 100-nanosecond intervals since January 1, 1601 (UTC).

2.2.3.5.11 Optional TLV 13: Error Codes

The Error Codes packet returns a set of error codes of type HRESULT.<13>



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

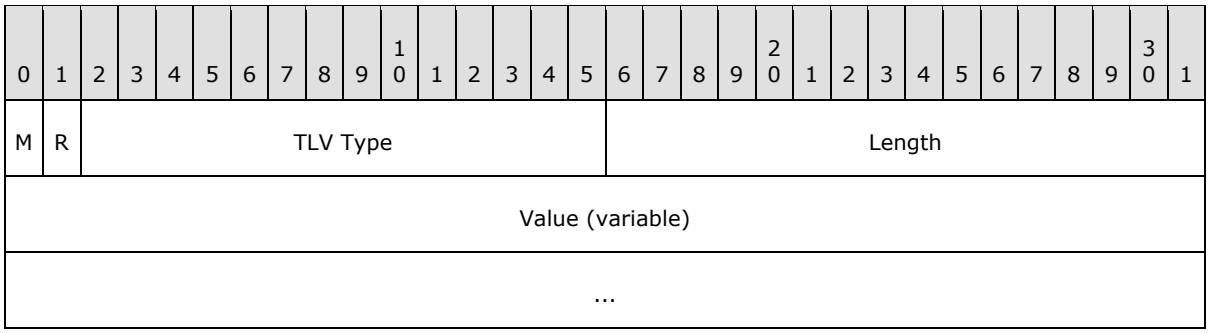
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 13.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): An array of HRESULTs.

2.2.3.5.12 Optional TLV 15: IPV6 Fix-up Servers

The IPV6 Fix-up Servers packet specifies the addresses of the IPV6 Fix-up Servers. An SoH MAY contain 0+ this attribute. An SoHR MAY contain 0 or 1 this attribute. A SHA may use this information to perform remediation.



M (1 bit): The **M** bit has the following possible values and MUST be set.

Value	Meaning
0	This is a non-mandatory TLV.
1	This is a mandatory TLV.

R (1 bit): The **R** bit is reserved, and MUST be set to zero and ignored on receipt.

TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 15.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): MUST be an array of 16-byte IPV6 addresses of the fix-up servers.

2.2.4 SSoHAttribute and SSoHRAttribute

The SSoHAttribute/SSoHRAttribute elements are elements that are used to construct valid SSoH and SSoHR TLVs. SSoHAttribute/SSoHRAttribute elements MUST be contained in the **Value** field of a TV (section 2.2.2).

When using a TV to contain an SSoH/SSoHR attribute, the TV types having values from 0 to 255 are reserved for use in the Statement of Health for NAP Protocol, and MUST NOT be used for other SSoH/SSoHR attributes. The following TV types are for use within this protocol:

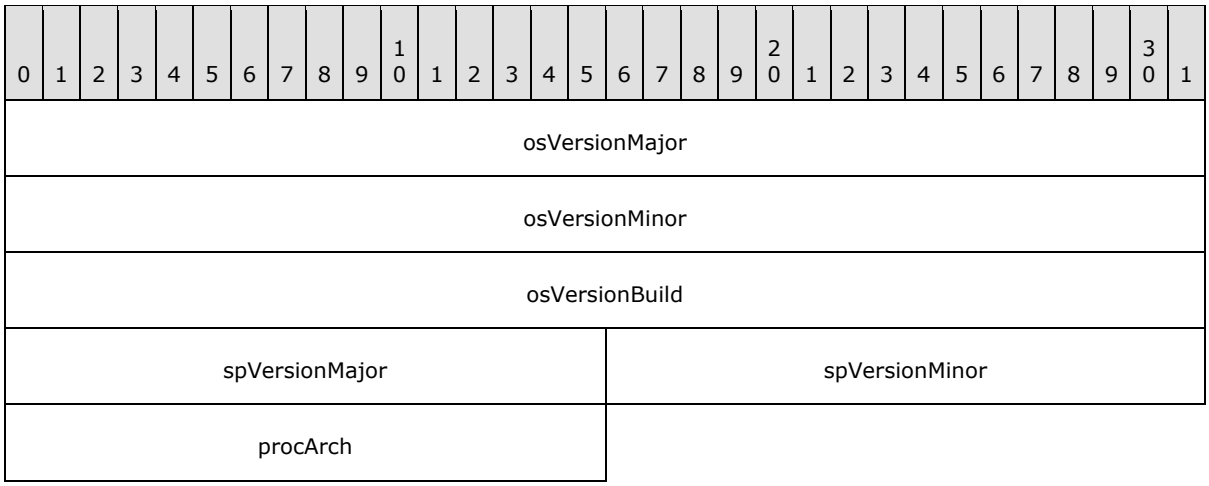
TV Type	Meaning
1	MS-Machine-Inventory
2	MS-Quarantine-State
3	MS-Packet-Info
4	MS-SystemGenerated-Ids
5	MS-MachineName
6	MS-CorrelationId
7	MS-Installed-Shvs
8	MS-Machine-Inventory-Ex

2.2.4.1 MS-Machine-Inventory Packet

The MS-Machine-Inventory attribute is used to communicate information about the host operating system and its processor architecture.<14>

The attribute MUST be present in an SSoH, and MAY be present in an SSoHR<15>.

The MS-Machine-Inventory packet:



osVersionMajor (4 bytes): A 32-bit unsigned integer that MUST specify the major version of the host operating system. Some examples are as follows:

Value	Meaning
0x00000004	The operating system is Windows NT 4.0.
0x00000005	The operating system is Windows Server 2003 R2, Windows Server 2003, Windows XP, or Windows 2000.
0x00000006	The operating system is Windows Vista or Windows Server 2008.

osVersionMinor (4 bytes): A 32-bit unsigned integer that MUST specify the minor version of the host operating system. Some examples are as follows:

Value	Meaning
0x00000000	The operating system is Windows Vista, Windows Server 2008, Windows 2000, or Windows NT 4.0.
0x00000001	The operating system is Windows XP.
0x00000002	The operating system is Windows Server 2003 R2, Windows Server 2003, or Windows XP Professional x64 Edition.

osVersionBuild (4 bytes): A 32-bit unsigned integer that MUST specify the build number of the host operating system.

spVersionMajor (2 bytes): A 16-bit unsigned integer that MUST specify the major version of the service pack installed on the host operating system. For example, for service pack 3, the major version number is 3. If no service pack has been installed, the value is zero.

spVersionMinor (2 bytes): A 16-bit unsigned integer that MUST specify the minor version of the service pack installed on the host operating system. For example, for service pack 3, the minor version number is 0.

procArch (2 bytes): A 16-bit unsigned integer that MUST specify the processor architecture of the host. Some examples are shown in the following table:

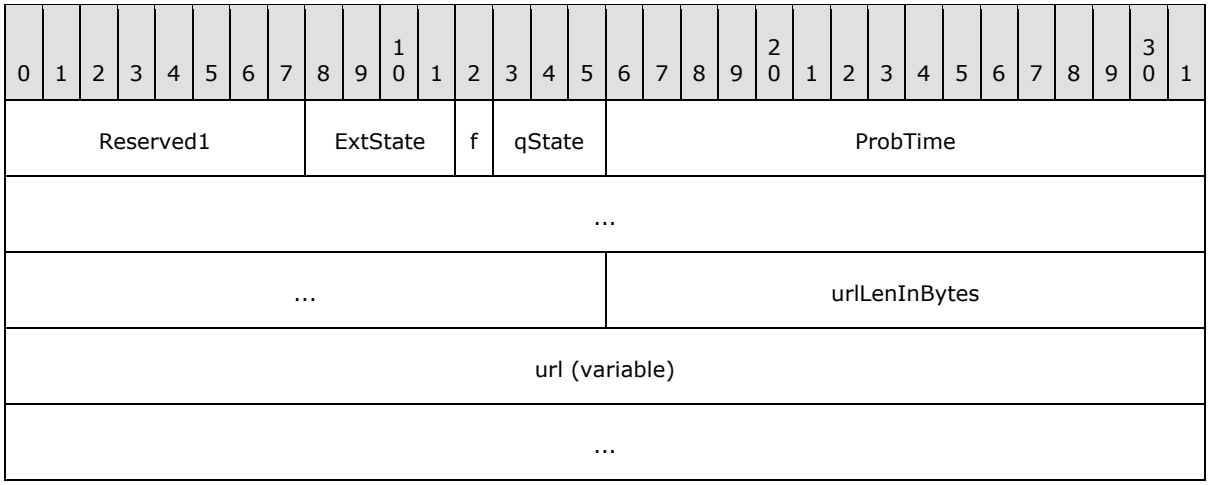
Value	Meaning
0x0000	x86 architecture.
0x0006	Intel Itanium Processor Family (IPF).
0x0009	x64 (AMD or Intel) architecture.
0xffff	Unknown processor.

2.2.4.2 MS-Quarantine-State Packet

The MS-Quarantine-State attribute is used to communicate information about the wanted or resulting permission to a requested network resource for a host. This attribute MUST be present in both the SSoH and the SSoHR.

The MS-Quarantine-State packet:

The first 16 bits is a field called **Flags**. This field contains the first four fields: **Reserved1**, **ExtState**, **f**, and **qState**.



Reserved1 (1 byte): 8-bit reserved field that MUST be set to zero and ignored on receipt.

ExtState (4 bits): 4-bit field that MUST have one of the following values:

Value	Meaning
0	No evaluation was done (this will be used if NPS policy does not return an extended state).
1	Machine is transitioning from one state to another.
2	Machine is infected, implying a bad health state.
3	Evaluation was done, but extended information could not be determined.

f (1 bit): 1-bit field indicating the Health Policy Server requires that the host MUST remediate any issues before attempting to access its resource again.

Value	Meaning
0	Remediation not required by policy.
1	Remediation required by policy.

qState (3 bits): 3-bit field that MUST be one of the following values:

Value	Meaning
1	Network connectivity is not being restricted.
2	Network connectivity is not being restricted but may be at a later time.
3	Network connectivity is being restricted.

ProbTime (8 bytes): A 64-bit field used to represent the time in which the client will be on probation. Probation allows an implementation to grant a client temporary authorization for a period even when the health check fails. At the end of the probation period, the client SHOULD revalidate its health. The behavior is implementation dependent and SHOULD be policy driven. The value MUST be formatted as a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).<16>

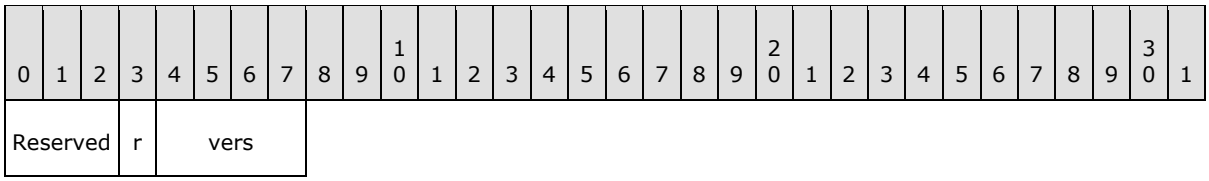
urlLenInBytes (2 bytes): A 16-bit field that MUST specify the length of the url field. The value of urlLenInBytes includes the NULL string termination character.

url (variable): UTF-8 (which MUST be as specified in [RFC2781]) representation of a **URL**.

2.2.4.3 MS-Packet-Info Packet

The MS-Packet-Info attribute is used to communicate information version and intent (request or response) of the SoH and SoHR. This attribute MUST be present in both the SSoH and the SSoHR.

The MS-Packet-Info packet:



Reserved (3 bits): The 3-bit Reserved Bits are reserved, MUST be set to zero and ignored on receipt.

r (1 bit): A 1-bit response/request flag. The value indicates if the attribute contains a request or a response. The field MUST contain one of the following values.

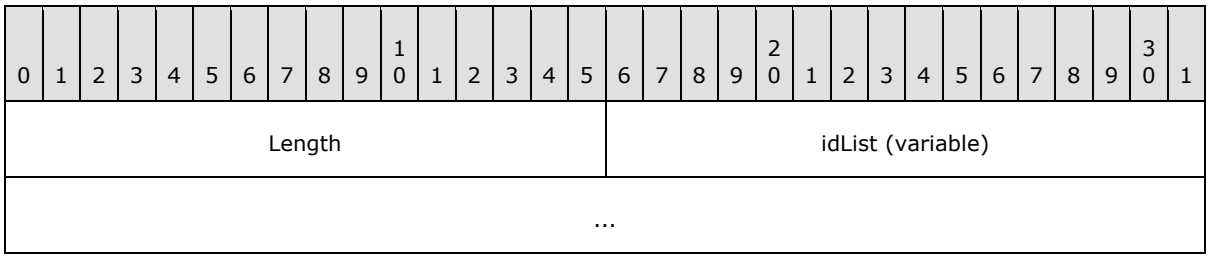
Value	Meaning
0	Response
1	Request

vers (4 bits): The 4-bit protocol version. MUST be set to 1. This is not to be confused with the version number set in the header.

2.2.4.4 MS-SystemGenerated-Ids Packet

The MS-SystemGenerated-Ids attribute contains a list of identifiers corresponding to SoHReportEntry values that contain error information as opposed to information about host state. This attribute MAY be present in an SSoH, and SHOULD NOT be present in an SSoHR.<17>

The MS-SystemGenerated-Ids packet:

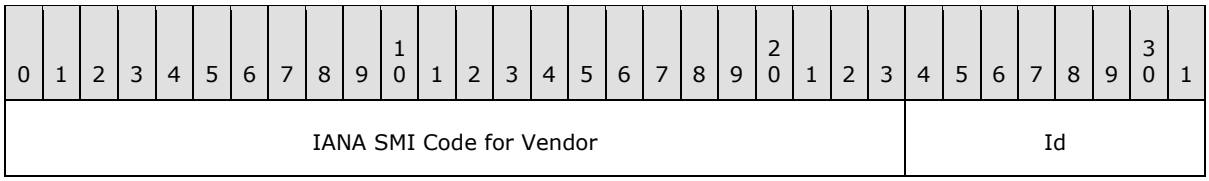


Length (2 bytes): 16-bit unsigned integer that MUST specify the length, in bytes, of the **idList** field.

idList (variable): MUST be an array of identifiers for the components that generated the SoHs/SoHRs in the message that contains this attribute. The identifiers MUST be formatted as specified in section 2.2.4.4.1.

2.2.4.4.1 MS-SystemGenerated-Ids Sub Packet

The MS-SystemGenerated-Ids Sub packet for the MS-SystemGenerated-Ids Packet idList field:



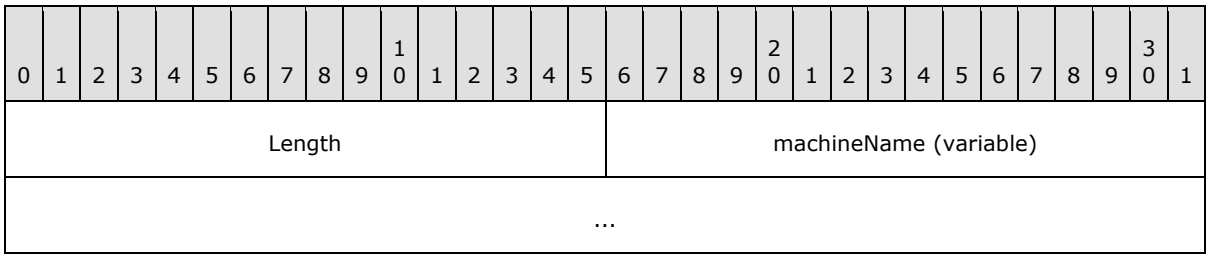
IANA SMI Code for Vendor (3 bytes): A 24-bit unsigned integer that MUST contain the IANA SMI code for the vendor whose component produced the message.

Id (1 byte): An 8-bit unsigned integer used to identify different components from the same vendor. Any value can be specified by the vendor for use by its components. This value, combined with the IANA SMI Code for Vendor, allows the routing of an SoHReportEntry from a client component to the corresponding server-side component that can deal with it. Similarly, the SoHRReportEntry is routed to the originator based on this ID. The IANA SMI Code for Vendor by itself is not sufficient because a given vendor's products may have multiple components. The 8-bit component identifier fully identifies the source and destination of each SoHReportEntry and SoHRReportEntry.

2.2.4.5 MS-MachineName Packet

The MS-MachineName attribute is used to communicate the name of the machine that generated the message. This attribute MUST be present in both the SSoH and the SSoHR.

The MS-MachineName packet:



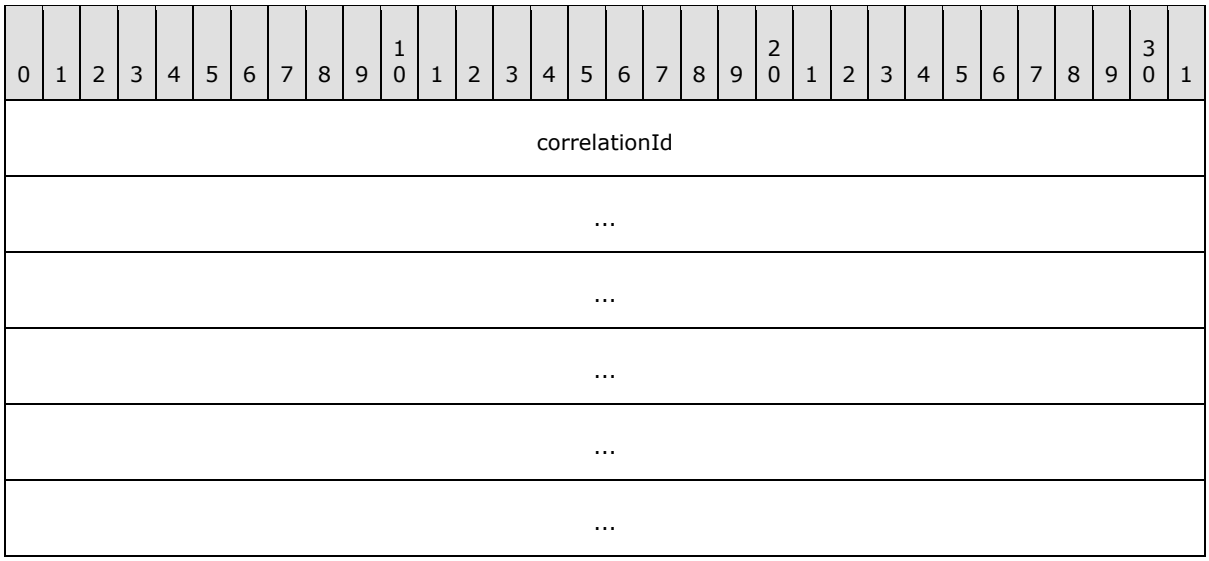
Length (2 bytes): A 16-bit field that MUST specify the length of the machineName field.

machineName (variable): A null-terminated UTF-8 encoded string field that MUST represent the **computer name** of the computer that generated the message.<18>

2.2.4.6 MS-CorrelationId Packet

The MS-CorrelationId attribute is used for diagnostic purposes to facilitate correlating messages related to a single transaction together. This attribute MUST be present in both the SSoH and the SSoHR.

The MS-CorrelationId packet:



correlationId (24 bytes): A 192-bit field that MUST represent a unique transaction identifier shared across SSoH and SSoHR messages. The format of the **correlationId** is implementation specific.<19>

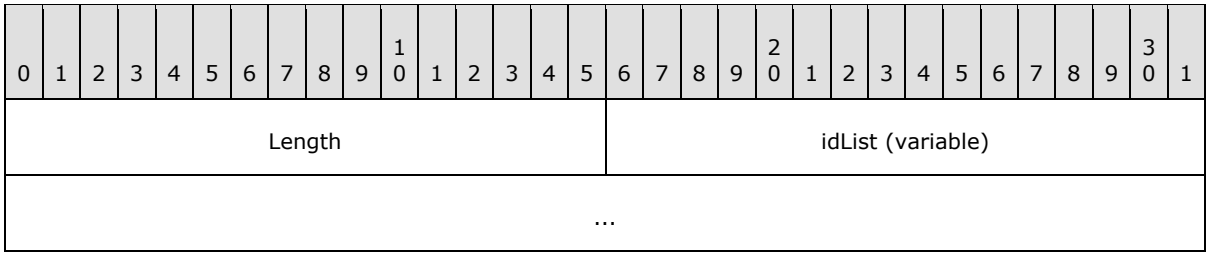
2.2.4.7 MS-Installed-Shvs Packet

The MS-Installed-Shvs is a list of identifiers of services that can evaluate SoHs on the SoH server.<20>

These identifiers of services can be used as hints to determine what **Health Messages** to send.<21>

This attribute SHOULD be present in the SSoHR and MAY be present in SSoH.

The MS-Installed-Shvs packet:

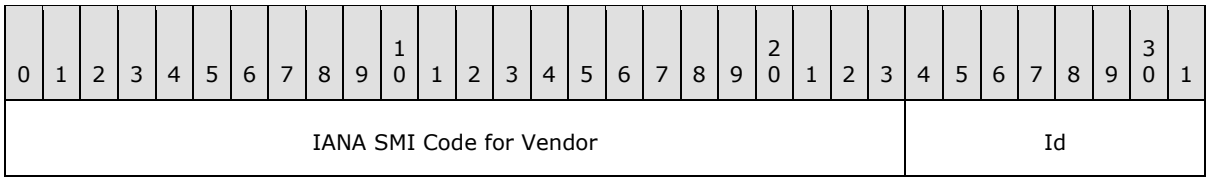


Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **idList** field.

idList (variable): MUST be an array of identifiers for the components that generated the SoHR messages on the server. The identifiers MUST be formatted as specified in section 2.2.4.7.1.

2.2.4.7.1 MS-Installed-Shvs Sub Packet

The MS-Installed-Shvs Sub packet for the MS-Installed-Shvs Packet idList field:



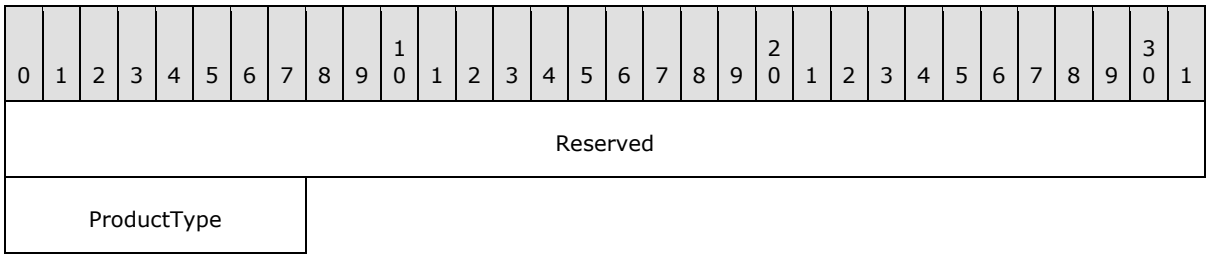
IANA SMI Code for Vendor (3 bytes): A 24-bit unsigned integer that MUST contain the IANA SMI code for the vendor whose component produced the message.

Id (1 byte): An 8-bit unsigned integer used to identify different components from the same vendor. Any value can be specified by the vendor for use by its components.

2.2.4.8 MS-Machine-Inventory-Ex Packet

The MS-Machine-Inventory-Ex packet is used to communicate additional information about the system sending the attribute. This attribute MUST be present in the SSoH and MAY be present in the SSoHR.<22>

The MS-Machine-Inventory-Ex packet:



Reserved (4 bytes): A 32-bit reserved value. MUST be ignored on receipt.

ProductType (1 byte): An 8-bit field used to represent the type of the operating system. It MUST have one of the following values:

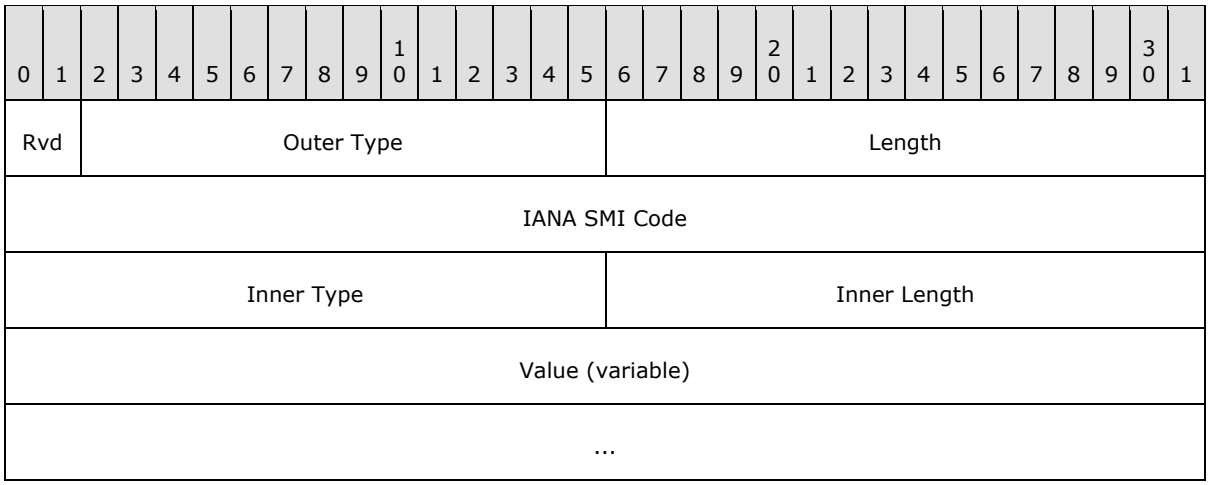
Value	Meaning
0x01	The system is a client.
0x02	The system is a domain controller running on a Windows server operating system
0x03	The system is a server.

2.2.5 SoH

The SoH message is used to represent a host's claims about its health state. It contains a header, the value field of which is the remainder of the message's content.

2.2.5.1 SoH Header

This is the SoH Header packet for the SoH:



Rvd (2 bits): The **Rvd** field is reserved, and MUST be set to zero and ignored on receipt.

Outer Type (14 bits): A 14-bit unsigned integer that MUST be set to 7.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the IANA SMI **Code** field, **Inner Type** field, **Inner Length** field and **Value** field.

IANA SMI Code (4 bytes): A 32-bit unsigned integer that MUST be set to 0x00000137 (see section 1.9). This value, in combination with the **Inner Type** value described below, allows implementations to identify that these messages belong to the Statement of Health for NAP Protocol. This is useful when implementations at either the client side or server side get other messages formatted similarly, as EAP TLVs.

Inner Type (2 bytes): A 16-bit unsigned integer that MUST have the value 0x0001 or 0x0002. This determines the version of the message content, and dictates the format of the data in the **Value** field.

Value	Meaning
0x0001	SSoH SoHReportEntry (0+)
0x0002	SoH Mode Sub-Header SSoH SoHReportEntry (0+)

Inner Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field.

Value (variable): A variable-length field that MUST contain data as follows:

If the **Inner Type** field is 0x0001:

- SSoH
- SoHReportEntry (0+)

If the **Inner Type** field is 0x0002:

- SoH Mode Sub-Header
- SSoH
- SoHReportEntry (0+)

2.2.5.2 SoHReportEntry

The SoHReportEntry message is used to represent a set of SoH attributes; it has no header of its own and is simply constructed as a set of SoH attributes.

The System-Health-ID SoH attribute MUST be the first SoH attribute. After this, any set of SoH attributes can be present (see section 2.2).

The SoHReportEntry message MUST contain one or more additional SoH attributes as follows (see section 2.2):

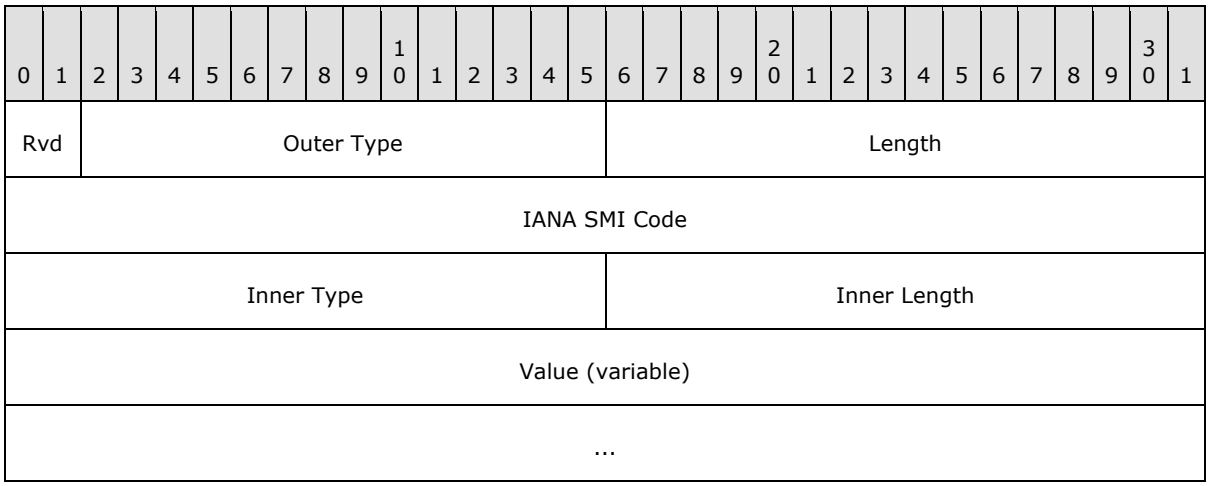
- System-Health-ID
- SoHAttribute (0+)

2.2.6 SoHR

The SoHR message is used to transport information about the result of the evaluation of an SoH message by the Health Policy Server. It contains a header, the value field of which is the rest of the message's content.

2.2.6.1 SoHR Header

This is the SoHR Header for the SoHR packet:



Rvd (2 bits): The **Rvd** field is reserved, and MUST be set to zero and ignored on receipt.

Outer Type (14 bits): A 14-bit unsigned integer that MUST be set to 7.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the IANA SMI **Code** field, **Inner Type** field, **Inner Length** field and **Value** field.

IANA SMI Code (4 bytes): A 32-bit unsigned integer that MUST be set to 0x00000137 (see section 1.9). This value, in combination with the **Inner Type** value described below, allows implementations to identify that these messages belong to the Statement of Health for NAP Protocol. This is useful when implementations at either the client side or server side get other messages formatted similarly, as EAP TLVs.

Inner Type (2 bytes): A 16-bit unsigned integer that MUST be set to 0x0001 or 0x0002. This value MUST be the same as the **Inner Type** value in the corresponding SoH message that the server received. This determines the version of the message content and dictates the format of the data in the **Value** field.

Value	Meaning
0x0001	SSoHR SoHRReportEntry (0+)
0x0002	SoH Mode Sub-Header SSoHR SoHRReportEntry (0+)

Inner Length (2 bytes): A 16-bit unsigned integer that MUST indicate the length, in bytes, of the **Value** field.

Value (variable): A variable-length field that MUST contain data as follows:

If the value of the **Inner Type** field is 0x0001:

- SSoHR
- SoHRReportEntry (0+)

If the value of the **Inner Type** field is 0x0002:

- SoH Mode Sub-Header
- SSoHR
- SoHRReportEntry (0+)

2.2.6.2 SoHRReportEntry

The SoHRReportEntry message is used to represent a set of attributes. It has no header of its own and is simply constructed as a set of SoHR attributes.

The System-Health-Id MUST be the first SoHR attribute.

The System-Health-Id MUST be followed by either a Compliance-Result-Codes, a Failure Category, or both.

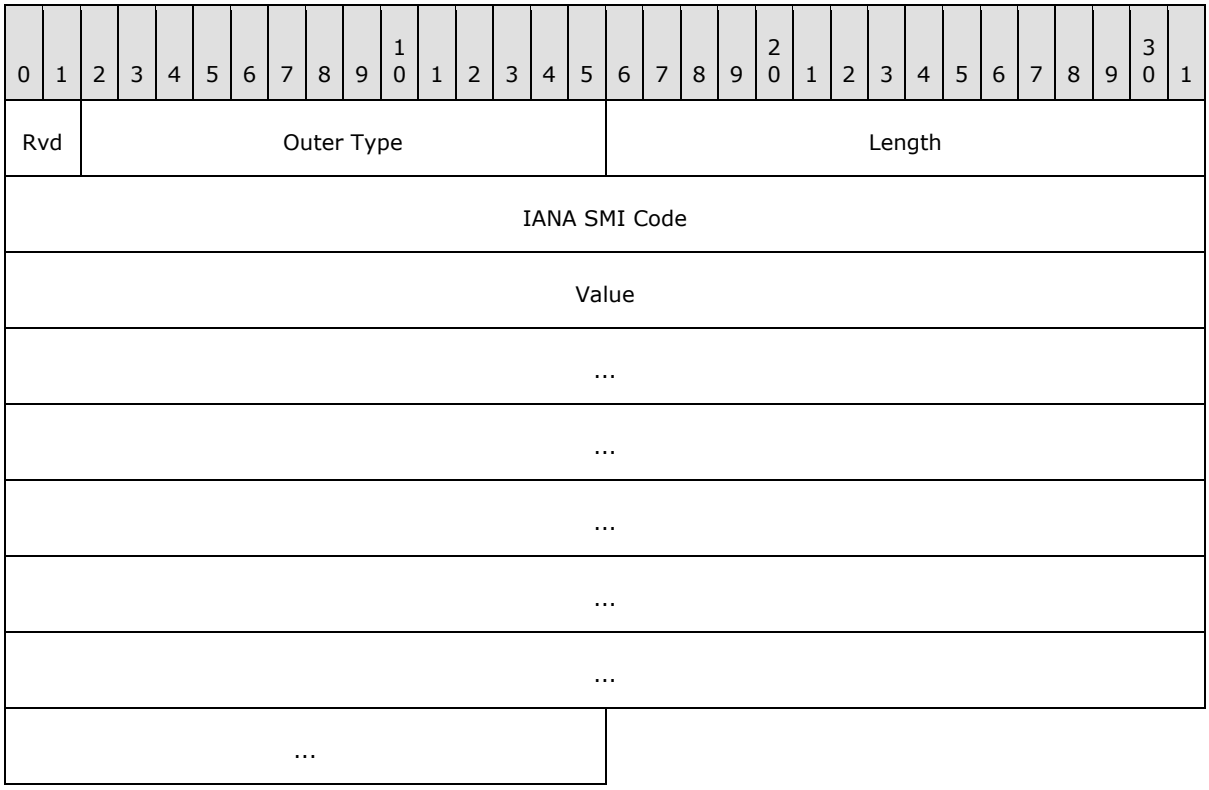
The SoHRReportEntry message MUST contain two or more additional SoHR attributes as follows (see section 2.2).

- System-Health-Id
- SoHRAttribute (1+) (MUST include a Compliance-Result-Codes, a Failure Category, or both.)

2.2.7 SoH Mode Sub-Header

The SoH mode sub-header is used to represent information about the information following it.

The SoH Mode Sub-Header packet:



Rvd (2 bits): The **Rvd** field is reserved, and MUST be set to zero and ignored on receipt.

Outer Type (14 bits): A 14-bit unsigned integer that MUST be set to 7.

Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the **Value** field and IANA SMI **Code**

IANA SMI Code (4 bytes): A 32-bit unsigned integer that MUST be set to 0x00000137 (see section 1.9).

Value (26 bytes): A field that MUST contain 26 bytes as follows:

Correlation ID: 24 bytes that MUST have the same value as the **Value** field of MS-CorrelationId.

Intent Flag: 1 byte that MUST be 0x01 for request (SoH message) and 0x00 for response (SoHR message).

Value	Meaning
0x01	SoH request message

Value	Meaning
0x00	SoHR response message

Content-Type Flag: 1 byte that MUST be 0x00. This field is intended to help in Statement of Health for NAP Protocol enhancements in the future.

2.2.8 SSoH

The SSoH message is used to represent generic information about the host, the message containing the SSoH, and the current health state of the host.

The SSoH message has no header of its own and is simply constructed as the following ordered sequence of SoH attributes and SSoH attributes:

- **System-Health-Id** attribute (see section 2.2.3.1). The value of this attribute MUST be decimal 79616 (0x00013700), created in accordance with the specifications in section 2.2.1.
- **Vendor-Specific** attribute (see section 2.2.3.3). The **Vendor-Specific** attribute MUST have the following fields set to the specified values:
  - **Vendor ID**: IANA SMI code, as specified in [IANA-ENT] set to 0x00000137.
  - **Value**: MUST contain the following SSoH attributes, see section 2.2.4:
    - **MS-Machine-Inventory** (see section 2.2.4.1).
    - **MS-Quarantine-State** (see section 2.2.4.2).
    - **MS-Packet-Info** (see section 2.2.4.3).
    - **MS-SystemGenerated-Ids** (optional; see section 2.2.4.4).
    - **MS-MachineName** (see section 2.2.4.5).
    - **MS-CorrelationId** (see section 2.2.4.6).
    - **MS-Machine-Inventory-Ex** (optional; see section 2.2.4.8).<23>

2.2.9 SSoHR

The SSoHR message is used to represent generic information about the Health Policy Server.

The SSoHR message has no header of its own and is simply constructed as the following ordered sequence of SoHR attributes and SSoHR attributes:

- **System-Health-Id** attribute (see section 2.2.3.1). The value of this attribute MUST be decimal 79616 (0x00013700), created in accordance to the specifications in section 2.2.1.
- **Vendor-Specific** attribute (see section 2.2.3.3). The **Vendor-Specific** attribute MUST have the following fields set to the specified values:
  - **Vendor ID**: IANA SMI code, as specified in [IANA-ENT] set to 0x00000137.
  - **Value**: MUST contain the following SSoHR attributes:
    - **MS-Packet-Info** (see section 2.2.4.3).

- **MS-MachineName** (see section 2.2.4.5).
- **MS-CorrelationId** (see section 2.2.4.6).
- **MS-Quarantine-State** (see section 2.2.4.2).
- **MS-Installed-Shvs** (optional; see section 2.2.4.7).<24>

3 Protocol Details

The following sections specify details of the Statement of Health for NAP Protocol, including abstract data models and message processing rules.

3.1 Common Details

3.1.1 Abstract Data Model

The abstract data models for client and server are specified in sections 3.2.1 and 3.3.1 respectively.

3.1.2 Timers

The Statement of Health for NAP Protocol includes no timers. The transports over which an SoH/SoHR is transported can have timers associated with them to achieve guaranteed and in-order delivery.

3.1.3 Initialization

The Statement of Health for NAP Protocol does not require explicit initialization. The transports that carry it may do so.

3.1.4 Higher-Layer Triggered Events

There are no common higher-layer triggered events.

3.1.5 Message Processing Events and Sequencing Rules

There are no common message processing events or sequencing rules.

3.1.6 Timer Events

There are no common timer events.

3.1.7 Other Local Events

There are no common local events.

3.2 Client-Specific Details

3.2.1 Abstract Data Model

The Statement of Health for NAP Protocol requires a single piece of state to be tracked on the SoH client:

MS-CorrelationId Cache: The SoH client MUST maintain a cache of the MS-CorrelationId values it sends. The cache is used to ensure that a received SoHR corresponds to an SoH that was sent.

Note The cache can be implemented using a variety of techniques. Any data structure that stores the above conceptual data may be used in the implementation.

3.2.2 Timers

More information is specified in section 3.1.2.

3.2.3 Initialization

More information is specified in section 3.1.3.

3.2.4 Higher-Layer Triggered Events

The following events can result in SoHs being sent by the SoH client:

1. A user reboots a machine.

2. The status of the client changes. For example, the firewall on the client is turned off.

In addition, events specific to the transport that carries the SoH messages can result in SoH messages being sent by the SoH client. For example, if DHCP is used to carry SoH messages, the renewal of the client IP address can result in an SoH message being sent to the server.

3.2.5 Message Processing Events and Sequencing Rules

The processing of SoH and SoHR messages on the client is implementation-specific and often involves the use of third-party components. An SoH contains data (contained in the SoHReportEntry values) that reports the client's current status to the health policy server. The value of this data is typically provided by software on the client that provides security services, such as an antivirus client or a security update client. Likewise, the validation of this data is typically provided by an antivirus server or a security update server. The Statement of Health for NAP Protocol itself simply provides the mechanism for this data to be supplied and evaluated.

An SoH client MUST do the following:

1. Create valid SoH messages containing host status information as per locally configured policy using one or more suitable SHAs.<25> The message version SHOULD be version 2.

2. Send the SoH messages over one or more suitable transports (for example, HCEP).<26>

3. Receive an SoHR over the transport that was used to send the SoH (for example, HCEP).<27>

4. Process the SoHRs received.<28>

3.2.5.1 Sending SoHs

An SoH client MUST ensure that all SoHs sent contain unique values in the MS-CorrelationId value of the SSoH included in the SoH.<29>

3.2.5.2 Receiving SoHs

An SoH client MUST discard any message received that is not a valid SoHR.

3.2.5.3 Sending SoHRs

An SoH client MUST NOT send an SoHR message.

3.2.5.4 Receiving SoHRs

The SoH client MUST ensure that every received SoHR is properly formed, including validating the length of each attribute. If the lengths are invalid, the SoH client MUST discard the SoHR message.

The SoH client MUST ensure that the SoHAttributes value in the SoHR contains at least a Compliance-Result-Codes or **Failure Category** attribute. If that is not the case, the SoH client MUST discard the SoHR message.

The SoH client MUST discard any received SoHR message that contains an MS-CorrelationId value in the SSoHR attribute that does not correspond to an MS-CorrelationId value previously sent in an SoH message.

3.2.6 Timer Events

Probation expiry timer event: Probation allows an implementation to grant a client temporary authorization for a period even when the health check fails. At the end of the probation period, the client SHOULD revalidate its health.

3.2.7 Other Local Events

The following events can result in SoHs being sent by the SoH client:

1. A new IP address is configured or assigned to the SoH client.
2. A new SoH transport protocol completes initialization.
3. A certificate expires.
4. A DHCP lease expires (for more information, see [RFC2131] section 4.4).
5. A DHCP lease renews (for more information, see [RFC2131] section 4.4).
6. 802.1x session authentication is started on the client.

3.3 Server-Specific Details

3.3.1 Abstract Data Model

No abstract data model is required.

3.3.2 Timers

More information is specified in section 3.1.2.

3.3.3 Initialization

More information is specified in section 3.1.3.

3.3.4 Higher-Layer Triggered Events

There are no higher-layer triggered events.

3.3.5 Message Processing Events and Sequencing Rules

The processing of SoH and SoHR messages on the server is implementation specific and often involves the use of SHVs, which can be developed by third parties. An SoHR contains data (contained in the SoHRReportEntry values) that reports the results of an evaluation of the client's current status. The value of this data is typically provided by other servers that provide security services, such as an antivirus server or a security update server. The Statement of Health for NAP

Protocol itself simply provides the mechanism for the client status to be supplied so that it can be evaluated.

A health policy server MUST do the following:

1. Receive and process SoHs.<30>
2. Create SoHRs.<31>
3. Send SoHRs.<32>

3.3.5.1 Sending SoHs

The health policy server MUST NOT send an SoH.

3.3.5.2 Receiving SoHs

The SoH server MUST ensure that every received SoH is properly formed, including validating the length of each attribute. The SoH server MUST then use the received SoH to evaluate the compliance of the SoH against policy.<33>

3.3.5.3 Sending SoHRs

The SoH server MUST ensure that every SSoHR it sends is properly formed, including validating the length of each attribute. The SoH server MUST include at least a valid Compliance-Result-Codes or **Failure Category** attribute in the **SoHRAttributeSet** of the SoHR.

The SoH server MUST NOT send an SoHR that is not in response to an SoH previously received. The SoH server MUST populate the value of the **MS-CorrelationId** attribute in the SoHR with the value of the **MS-CorrelationId** attribute in the SoH to which this SoHR is a response.

3.3.5.4 Receiving SoHRs

The SoH server MUST discard any message that is not a valid SoH.

3.3.6 Timer Events

There are no timer events on the SoH server.

3.3.7 Other Local Events

The following event MUST result in an SoHR being sent by the SoH server:

- A RADIUS packet containing a Statement of Health for NAP Protocol attribute is received.

4 Protocol Examples

This is a simple protocol with a single exchange. The party seeking access to a network resource sends the SoH and receives an SoHR. It is represented graphically below.

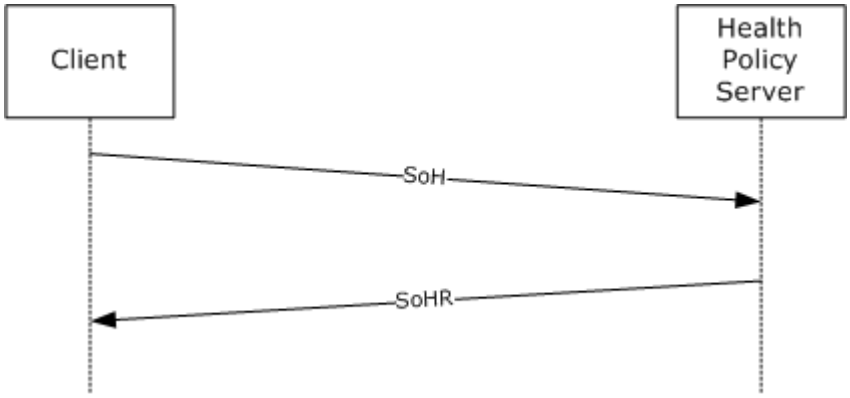


Figure 3: Client SoH request and health policy server response

In all cases, a transport protocol is involved in sending the messages in both directions. The transport protocol is typically the authentication protocol that mediates access to the network resource. This simple flow applies to all use cases. For specifics about the flow, see section 3.

When the SoH is being sent, it is likely that the client is requesting access to some service and is being required to prove good health as a precondition. When the SoH is received, it is likely that the receiver will forward it to some infrastructure server that will evaluate the SoH and return the response (or SoHR) to the client via the original receiver of the SoH.

The receipt of an SoHR by the client, generally, allows access to the service being requested. In cases when the health of the client is not good, the SoHR is likely to contain sufficient instructions to permit the client to seek and receive remedy. Once the client is remedied, it can initiate the protocol again, this time in good health.

5 Security

The following sections specify security considerations for implementers of the Statement of Health for NAP Protocol.

5.1 Security Considerations for Implementers

Security for health messages should be provided by the transport layer protocol. The transport protocol should guard against replay and tampering, and provide privacy of health messages. Health messages should not be transmitted unencrypted even if the transport protocol itself does not encrypt the communication. In such cases, the individual messages should be encrypted, signed, and time stamped to ensure their integrity and confidentiality, and to prevent usage of an SoH after it no longer represents the state of the computer.

Version 2 of the protocol may support enhancements in a later version. Even though version 2 does not offer additional security over version 1 of the Statement of Health for NAP Protocol, implementers should use it for future compatibility and security enhancements.

Implementers may use the protocol in applications where the messages are carried in a transport that does not provide security (for example, DHCP). In such cases, it is important for them to guide their users (the network administrators) to have infrastructure measures in place that accommodate and compensate for such usage.

The following risks are mitigated when the transport provides security for health messages:

1. Confidentiality from Passive Observation. Health messages contain information that may not only disclose personally identifying information of a user but also disclose a current security issue in the system. That is the nature of the message and the service it provides. For this reason, it is important to preserve the confidentiality of these messages. It should not be possible for a **man-in-the-middle (MITM) attack** to successfully view the contents of health messages as they are transmitted.

2. Spoofing. Health message (an SoH) is a token that potentially causes a client to be authorized to access a protected resource. It should not be possible for anyone other than the system that created the SoH to use it. This requires that the authenticity of the source be verified. Similarly, an SoHR potentially causes a client to execute code that may be unsafe. For this reason, it is important to prevent an attacker from being able to spoof such messages. Thus, it should not be possible for an attacker to impersonate a network access server (NAS), **EAP server**, or a client.

3. Active Tampering. For the risks discussed above, it should not be possible for an attacker to modify the SoHR undetected. Similarly, tampering of the SoH may cause a client to be given access when it must not, and vice versa. The security provided by the transport mechanism should prevent tampering with these messages.

4. Replaying. A message that causes a client to be granted access can potentially be retransmitted by another client to incorrectly give it access. This should be prevented by ensuring **idempotence** of SoH and SoHR messages as observed by a man in the middle who is able to view the transport-level communication.

There are a set of attacks for which no effective measures currently exist. These are documented here to make developers aware of them.

There are no reliable measures that can prevent a denial of service attack on either a client or a NAS. Such attacks may include network flooding or tampering of communications by an attacker who is on the path between a client and a NAS (for example, in the case of WiFi).

There is always the potential that the host itself is compromised by some kernel mode malicious software (malware). In such cases, the SoHs and SoHAttributes produced by the client cannot be trusted. However, there are no effective solutions for this currently, absent broad deployment of trusted hardware.

5.2 Index of Security Parameters

There are no security parameters for this protocol.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2008
- Windows Vista

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1.5: Client and server prerequisites for Network Access Protection: The Windows implementation of the SoH Client sends version 2 messages by default. It can be configured to send version 1 messages. The Windows implementation of the SoH server accepts versions 1 and 2 messages.

<2> Section 1.5: Client prerequisites for Network Access Protection: The client needs to be configured to know what access methods are required to enable the Network Access Protection solution.

<3> Section 2.2.3.1: Statement of Health for NAP Protocol IDs that are used. The following IDs are used in the Windows implementation of the Statement of Health for NAP Protocol: 0x00013700; 0x00013701; 0x00013702; 0x00013703; 0x00013704; 0x00013705; 0x00013706; 0x00013707; 0x00013780; 0x00013781.

<4> Section 2.2.3.3: An IANA-assigned SMI vendor ID is strongly recommended. Other types of IDs are also acceptable as long as they can uniquely identify the vendor that specifies the data in the Data field. For example in the Microsoft Windows implementation, Windows Security Health Agent (WSHA) uses the following vendor ID format: IANA-assigned SMI vendor code + component ID (same as the Health ID format specified in Section 2.2.3.1).

<5> Section 2.2.3.5: Third parties are allowed to use these optional TLVs in their implementations to construct their own SoH/SoHR messages. The Windows implementation does not check Value fields of the optional TLVs in the third-parties implementations. However, the Microsoft Windows implementation does perform consistency checks on the length of the attributes. For example, if Optional TLV 5 (the Time-of-Last-Update TLV) is used in a third-party SoH/SoHR message, the Windows implementation requires that the length of its **Value** field be 8, but does not care what the specific contents of the **Value** field is.

<6> Section 2.2.3.5.4: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<7> Section 2.2.3.5.5: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<8> Section 2.2.3.5.6: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<9> Section 2.2.3.5.7: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<10> Section 2.2.3.5.8: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<11> Section 2.2.3.5.9: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<12> Section 2.2.3.5.10: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<13> Section 2.2.3.5.11: Third-parties SHA/SHV implementations and MSSHA/SHV may use 0+ this optional TLV in their SoH/SoHR messages. The Windows implementation of the SoH protocol does not care when or how many of these optional TLV's are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in WSHA SoH and WSHV SoHR [MS-WSH].

<14> Section 2.2.4.1: MS-Machine-Inventory (section 2.2.4.1) fields that are populated. Windows populates the **osVersionMajor**, **osVersionMinor**, **spVersionMajor**, **spVersionMinor**, and **procArch** fields based on the returns from GetVersionInfoEx and its OSVERSIONINFOEX structure (for more information, see [MSDN-OSVERSIONINFOEX]).

<15> Section 2.2.4.1: The health policy server does not send this TLV in the SSoHR. The client does not expect nor read this TLV when the SSoHR is parsed.

<16> Section 2.2.4.2: Client end-of-probation time resubmit. At the end of the probation time, the client resubmits itself for validation. The resubmission process depends on how the client is deployed. For example, in the case of an HRA deployment, the client has its certificate expiry at the end of probation time and enrolls for a new certificate by posting a new HCEP message at that time. When the client resubmits, Windows evaluates its compliance and grants access according to policy.

<17> Section 2.2.4.4: MS-SystemGenerated-Ids attribute for internal error indication: The Windows client includes the MS-SystemGenerated-Ids attribute when an internal error occurs while attempting to gather state information about the host. The Windows server does not include this attribute.

<18> Section 2.2.4.5: The MS-MachineName (section 2.2.4.5) attribute used to submit the name: The **machineName** attribute is the fully qualified domain name (FQDN) of the computer if joined to a Windows domain; otherwise, the computer name is used.

<19> Section 2.2.4.6: The MS-CorrelationId (section 2.2.4.6) for diagnostic purposes: The **correlationId** is a concatenation of the 16-byte connection ID and the **FILETIME** at which the attribute was generated.

<20> Section 2.2.4.7: The SoH Evaluation API: Implemented in Microsoft Windows Server 2008 only. The Health Policy Server in Windows includes an API that enables plug-ins that perform SoH evaluation to register. The Windows SoH server only includes the MS-Installed-Shvs (section 2.2.4.7) attribute if such plug-ins are registered.

<21> Section 2.2.4.7: The Windows client always sends all available Health Messages.

<22> Section 2.2.4.8: The health policy server does not send this TLV in the SSoHR. The client does not expect nor read this TLV when the SSoHR is parsed.

<23> Section 2.2.8: MS-SystemGenerated-Ids is a list of component-ids that is unable to provide an SoHReportEntry. The Windows implementation of the Statement of Health for NAP Protocol includes the MS-SystemGenerated-Ids with a list of component-ids that are unable to provide an SoHReportEntry at the time the SSoH is generated. It always sends the optional MS-Machine-Inventory-Ex attribute in the SSoH.

<24> Section 2.2.9: MS-Installed-Shvs has a list of SHVs that are installed. Implemented in Windows Server 2008 only. The Windows implementation of the health policy server includes the MS-Installed-Shvs with a list of SHVs that are installed on the server to perform health validation of SoHReportEntry.

<25> Section 3.2.5: The SoH client includes an application programming interface (API) to allow plug-ins to report client state. The WindowsSoH client, also called the Network Access Protection (NAP) agent, includes an application programming interface (API) that allows plug-ins that report client state to register with the system. These plug-ins are called system health agents (SHAs). Example SHAs include antivirus and security update clients. One specific SHA, Windows Security Health Agent (WSHA), as specified in [MS-WSH], is included in Windows. Each SHA produces an SoHReportEntry for the state it reports. The NAP agent forms an SoH by appending the collection of the SoHReportEntry from the SHAs to a valid SSoH. The NAP agent creates a new SoH whenever it receives a new SoHReportEntry from an SHA, or whenever a QEC requests one.

<26> Section 3.2.5: The NAP agent allows plug-ins to register. These plug-ins are called quarantine enforcement clients (QEC). The NAP agent includes an application programming interface that allows plug-ins that transport SoHs to register with the system. These plug-ins are called quarantine enforcement clients (QEC). Windows includes QECs for transporting SoHs over HCEP, HTTP, DHCP, and EAP. A QEC transports an SoH whenever it receives one from the NAP agent if the QEC has been enabled by the administrator.

<27> Section 3.2.5: QECs receive SoHRs and then pass the SoHRs to the NAP agent. When the Windows QECs receive SoHRs in response to the SoHs they transport, the QECs can then pass the SoHRs to the NAP agent for processing.

<28> Section 3.2.5: The NAP agent validates the SoHR and notifies the user and SHA/SoHR. When the NAP agent receives the SoHR, it validates it and notifies the user if the MS-Quarantine-State value indicates that the user's network connectivity is restricted. The NAP agent delivers the SoHReportEntry values in the SoHR to the appropriate SHA for processing according to the System-Health-ID attribute in the SoHRReportEntry.

<29> Section 3.2.5.1: MS-CorrelationId creates a GUID corresponding to the network connection. The NAP agent forms its MS-CorrelationId by filling in the first 16 bytes with the value of a GUID corresponding to the network connection over which the SoH is being transported. These 16 bytes are the same each time the SoH is transported over that connection. The last 8 bytes of the MS-CorrelationId is a 64-bit unsigned integer representing the number of 100-nanosecond intervals between January 1, 1601 (UTC) and when the SoH was delivered for transport.

<30> Section 3.3.5: The Windows health policy server is part of the network policy server (NPS). This is implemented in Windows Server 2008 only. NPS is the Windows RADIUS server. It includes an application programming interface that allows plug-ins to validate the SoHReportEntry messages sent inside an SoH by SoH clients. These plug-ins are called system health validators (SHV). One specific SHV, Windows Security Health Validator (WSHV), as specified in [MS-WSH], is included in Windows. The health policy server validates the format of a received SoH and delivers the

SoHReportEntry values to the appropriate SHV for evaluation based on the value of the **SystemHealthId** attribute in the SoHReportEntry. It then waits for the SHVs to complete their evaluation of the SoHReportEntry values.

<31> Section 3.3.5: SHVs evaluate the SoHReportEntry values by delivering SoHReportEntry values to the health policy server. This is implemented in Windows Server 2008 only. The SHVs complete their evaluation of the SoHReportEntry values by delivering SoHReportEntry values to the health policy server. The health policy server forms an SSoHR and populates the **Quarantine-State** attribute therein, according to policy taking the Compliance-Result-Codes and **Failure Category** attributes as input. The health policy server forms a valid SoHR using the resulting SSoHR and the collection of SoHReportEntry messages received from the SHVs.

<32> Section 3.3.5: The health policy server delivers the SoH to NPS to be processed against policy, and sends it via RADIUS to the SoH client. This is implemented in Windows Server 2008 only. The health policy server delivers the SoHR to NPS, which processes it against policy and sends it as a vendor-specific attribute (VSA) in RADIUS to a RADIUS client, as specified in [MS-RNAP]. The RADIUS client then delivers the SoHR to the SoH client over the transport that was originally used to send the SoH. An example of this process is specified in [MS-HCEP] section 1.3.

<33> Section 3.3.5.2: The server SoH evaluation is implemented in Windows Server 2008 only. The process Windows Server 2008 uses to evaluate the SoH is specified in section 3.2.

7 Index

S

T

V

W