

Text Comparison

Documents Compared

MS-SOH.5.0.1.pdf

MS-SOH.5.0.2.pdf

Summary

977 word(s) added

504 word(s) deleted

17245 word(s) matched

106 block(s) matched

To see where the changes are, scroll down.

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

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. You may also distribute in your implementation any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary

Date	Revision History	Revision Class	Comments
03/14/2007	1.0	Major	Updated and revised the technical content.
04/10/2007	1.1	Minor	Updated the technical content.
05/18/2007	1.2	Minor	Addressed EU feedback

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

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. You may also distribute in your implementation any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary

Date	Revision History	Revision Class	Comments
03/14/2007	1.0	Major	Updated and revised the technical content.
04/10/2007	1.1	Minor	Updated the technical content.
05/18/2007	1.2	Minor	Addressed EU feedback

Date	Revision History	Revision Class	Comments
06/08/2007	1.2.1	Editorial	Revised and edited the technical content.
07/10/2007	1.2.2	Editorial	Revised and edited the technical content.
08/17/2007	1.2.3	Editorial	Revised and edited the technical content.
09/21/2007	2.0	Major	Updated and revised the technical content.
10/26/2007	2.0.1	Editorial	Revised and edited the technical content.
01/25/2008	3.0	Major	Updated and revised the technical content.
03/14/2008	3.0.1	Editorial	Revised and edited the technical content.
06/20/2008	4.0	Major	Updated and revised the technical content.
07/25/2008	4.0.1	Editorial	Revised and edited the technical content.
08/29/2008	4.0.2	Editorial	Revised and edited the technical content.
10/24/2008	4.0.3	Editorial	Revised and edited the technical content.
12/05/2008	5.0	Major	Updated and revised the technical content.
01/16/2009	5.0.1	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
06/08/2007	1.2.1	Editorial	Revised and edited the technical content.
07/10/2007	1.2.2	Editorial	Revised and edited the technical content.
08/17/2007	1.2.3	Editorial	Revised and edited the technical content.
09/21/2007	2.0	Major	Updated and revised the technical content.
10/26/2007	2.0.1	Editorial	Revised and edited the technical content.
01/25/2008	3.0	Major	Updated and revised the technical content.
03/14/2008	3.0.1	Editorial	Revised and edited the technical content.
06/20/2008	4.0	Major	Updated and revised the technical content.
07/25/2008	4.0.1	Editorial	Revised and edited the technical content.
08/29/2008	4.0.2	Editorial	Revised and edited the technical content.
10/24/2008	4.0.3	Editorial	Revised and edited the technical content.
12/05/2008	5.0	Major	Updated and revised the technical content.
01/16/2009	5.0.1	Editorial	Revised and edited the technical content.
<u>02/27/2009</u>	<u>5.0.2</u>	<u>Editorial</u>	<u>Revised and edited the technical content.</u>

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Protocol Overview (Synopsis)	7
1.4 Relationship to Other Protocols	8
1.5 Prerequisites/Preconditions	8
1.6 Applicability Statement	9
1.7 Versioning and Capability Negotiation	9
1.8 Vendor-Extensible Fields	9
1.9 Standards Assignments	10
2 Messages	11
2.1 Transport	11
2.2 Message Syntax	11
2.2.1 Type-Length-Value (TLV) Packet	12
2.2.2 Type-Value (TV) Packet	13
2.2.3 SoHAttributes / SoHRAAttributes	13
2.2.3.1 System-Health-ID Packet	13
2.2.3.2 Compliance-Result-Codes	14
2.2.3.3 Vendor-Specific Packet	15
2.2.3.4 Failure Category	16
2.2.3.5 Optional TLVs	16
2.2.3.5.1 Optional TLV 0: Reserved	17
2.2.3.5.2 Optional TLV 1: Reserved	18
2.2.3.5.3 Optional TLV 3: IPv4 Fix-up Servers	19
2.2.3.5.4 Optional TLV 5: Time-of-Last-Update	19
2.2.3.5.5 Optional TLV 6: Client-ID	20
2.2.3.5.6 Optional TLV 8: Health-Class	21
2.2.3.5.7 Optional TLV 9: Software-Version	21
2.2.3.5.8 Optional TLV 10: Product-Name	22
2.2.3.5.9 Optional TLV 11: Health Class Status	22
2.2.3.5.10 Optional TLV 12: SOH Generation Time	23
2.2.3.5.11 Optional TLV 13: Error Codes	24
2.2.3.5.12 Optional TLV 15: IPv6 Fix-up Servers	24
2.2.4 SSoHAttribute and SSoHRAAttribute	25
2.2.4.1 MS-Machine-Inventory Packet	25
2.2.4.2 MS-Quarantine-State Packet	27
2.2.4.3 MS-Packet-Info Packet	28
2.2.4.4 MS-SystemGenerated-Ids Packet	29
2.2.4.4.1 MS-SystemGenerated-Ids Subpacket	29
2.2.4.5 MS-MachineName Packet	29
2.2.4.6 MS-CorrelationId Packet	30
2.2.4.7 MS-Installed-Shvs Packet	30
2.2.4.7.1 MS-Installed-Shvs Subpacket	31
2.2.4.8 MS-Machine-Inventory-Ex Packet	31
2.2.5 SoH	32
2.2.5.1 SoH Header	32
2.2.5.2 SoHReportEntry	33

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Protocol Overview (Synopsis)	7
1.4 Relationship to Other Protocols	8
1.5 Prerequisites/Preconditions	8
1.6 Applicability Statement	9
1.7 Versioning and Capability Negotiation	9
1.8 Vendor-Extensible Fields	9
1.9 Standards Assignments	10
2 Messages	11
2.1 Transport	11
2.2 Message Syntax	11
2.2.1 Type-Length-Value (TLV) Packet	12
2.2.2 Type-Value (TV) Packet	13
2.2.3 SoHAttributes / SoHRAAttributes	13
2.2.3.1 System-Health-ID Packet	13
2.2.3.2 Compliance-Result-Codes	14
2.2.3.3 Vendor-Specific Packet	15
2.2.3.4 Failure Category	16
2.2.3.5 Optional TLVs	16
2.2.3.5.1 Optional TLV 0: Reserved	17
2.2.3.5.2 Optional TLV 1: Reserved	18
2.2.3.5.3 Optional TLV 3: IPv4 Fix-up Servers	19
2.2.3.5.4 Optional TLV 5: Time-of-Last-Update	19
2.2.3.5.5 Optional TLV 6: Client-ID	20
2.2.3.5.6 Optional TLV 8: Health-Class	21
2.2.3.5.7 Optional TLV 9: Software-Version	21
2.2.3.5.8 Optional TLV 10: Product-Name	22
2.2.3.5.9 Optional TLV 11: Health Class Status	22
2.2.3.5.10 Optional TLV 12: SOH Generation Time	23
2.2.3.5.11 Optional TLV 13: Error Codes	24
2.2.3.5.12 Optional TLV 15: IPv6 Fix-up Servers	24
2.2.4 SSoHAttribute and SSoHRAAttribute	25
2.2.4.1 MS-Machine-Inventory Packet	25
2.2.4.2 MS-Quarantine-State Packet	27
2.2.4.3 MS-Packet-Info Packet	28
2.2.4.4 MS-SystemGenerated-Ids Packet	29
2.2.4.4.1 MS-SystemGenerated-Ids Subpacket	29
2.2.4.5 MS-MachineName Packet	29
2.2.4.6 MS-CorrelationId Packet	30
2.2.4.7 MS-Installed-Shvs Packet	30
2.2.4.7.1 MS-Installed-Shvs Subpacket	31
2.2.4.8 MS-Machine-Inventory-Ex Packet	31
2.2.5 SoH	32
2.2.5.1 SoH Header	32
2.2.5.2 SoHReportEntry	33

2.2.6	SoHR	33
2.2.6.1	SoHR Header	33
2.2.6.2	SoHRReportEntry	35
2.2.7	SoH Mode Subheader	35
2.2.8	SSoH	36
2.2.9	SSoHR	37
3	Protocol Details	38
3.1	Common Details	38
3.1.1	Abstract Data Model	38
3.1.2	Timers	38
3.1.3	Initialization	38
3.1.4	Higher-Layer Triggered Events	38
3.1.5	Message Processing Events and Sequencing Rules	38
3.1.6	Timer Events	38
3.1.7	Other Local Events	38
3.2	Client-Specific Details	38
3.2.1	Abstract Data Model	38
3.2.2	Timers	39
3.2.3	Initialization	39
3.2.4	Higher-Layer Triggered Events	39
3.2.5	Message Processing Events and Sequencing Rules	39
3.2.5.1	Sending SoHs	39
3.2.5.2	Receiving SoHs	39
3.2.5.3	Sending SoHRs	39
3.2.5.4	Receiving SoHRs	40
3.2.6	Timer Events	40
3.2.7	Other Local Events	40
3.3	Server-Specific Details	40
3.3.1	Abstract Data Model	40
3.3.2	Timers	40
3.3.3	Initialization	40
3.3.4	Higher-Layer Triggered Events	40
3.3.5	Message Processing Events and Sequencing Rules	41
3.3.5.1	Sending SoHs	41
3.3.5.2	Receiving SoHs	41
3.3.5.3	Sending SoHRs	41
3.3.5.4	Receiving SoHRs	41
3.3.6	Timer Events	41
3.3.7	Other Local Events	41
4	Protocol Examples	42
5	Security	43
5.1	Security Considerations for Implementers	43
5.2	Index of Security Parameters	44
6	Appendix A: Windows Behavior	45
7	Change Tracking	49
8	Index	50

2.2.6	SoHR	33
2.2.6.1	SoHR Header	33
2.2.6.2	SoHRReportEntry	35
2.2.7	SoH Mode Subheader	35
2.2.8	SSoH	36
2.2.9	SSoHR	37
3	Protocol Details	38
3.1	Common Details	38
3.1.1	Abstract Data Model	38
3.1.2	Timers	38
3.1.3	Initialization	38
3.1.4	Higher-Layer Triggered Events	38
3.1.5	Message Processing Events and Sequencing Rules	38
3.1.6	Timer Events	38
3.1.7	Other Local Events	38
3.2	Client-Specific Details	38
3.2.1	Abstract Data Model	38
3.2.2	Timers	39
3.2.3	Initialization	39
3.2.4	Higher-Layer Triggered Events	39
3.2.5	Message Processing Events and Sequencing Rules	39
3.2.5.1	Sending SoHs	39
3.2.5.2	Receiving SoHs	39
3.2.5.3	Sending SoHRs	39
3.2.5.4	Receiving SoHRs	40
3.2.6	Timer Events	40
3.2.7	Other Local Events	40
3.3	Server-Specific Details	40
3.3.1	Abstract Data Model	40
3.3.2	Timers	40
3.3.3	Initialization	40
3.3.4	Higher-Layer Triggered Events	40
3.3.5	Message Processing Events and Sequencing Rules	41
3.3.5.1	Sending SoHs	41
3.3.5.2	Receiving SoHs	41
3.3.5.3	Sending SoHRs	41
3.3.5.4	Receiving SoHRs	41
3.3.6	Timer Events	41
3.3.7	Other Local Events	41
4	Protocol Examples	42
5	Security	43
5.1	Security Considerations for Implementers	43
5.2	Index of Security Parameters	44
6	Appendix A: Windows Behavior	45
7	Change Tracking	49
8	Index	51

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 also 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 measures for security health and system health.

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 type-length-value
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

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 the 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, then no differences, no errors, and no inconsistencies will result. Example: $\text{abs}(x) == \text{abs}(\text{abs}(x)) == \text{abs}(\text{abs}(\text{abs}(x))) == \dots$ for all x .

man in the middle attack (MITM): 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 **MITM** attack, the users are unaware that there is

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 also 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 measures for security health and system health.

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 type-length-value
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

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 the 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, then no differences, no errors, and no inconsistencies will result. Example: $\text{abs}(x) == \text{abs}(\text{abs}(x)) == \text{abs}(\text{abs}(\text{abs}(x))) == \dots$ for all x .

man in the middle attack (MITM): 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 **MITM** attack, the users are unaware that there is

an attacker, which is intercepting and modifying their data, between them. 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 length. 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](#)", March 2007.

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

[MS-HCEP] Microsoft Corporation, "[Health Certificate Enrollment Protocol Specification](#)", July 2006.

[MS-RNAP] Microsoft Corporation, "[Vendor-Specific RADIUS Attributes for Network Access Protection \(NAP\) Data Structure](#)", July 2006.

an attacker, which is intercepting and modifying their data, between them. 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 length. 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](#)", March 2007.

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

[MS-HCEP] Microsoft Corporation, "[Health Certificate Enrollment Protocol Specification](#)", July 2006.

[MS-RNAP] Microsoft Corporation, "[Vendor-Specific RADIUS Attributes for Network Access Protection \(NAP\) Data Structure](#)", July 2006.

[MS-WSH] Microsoft Corporation, "[Windows Security Health Agent \(WSHA\) and Windows Security Health Validator \(WSHV\) Protocol Specification](#)", April 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](#)", July 2006.

[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 in which 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.

[MS-WSH] Microsoft Corporation, "[Windows Security Health Agent \(WSHA\) and Windows Security Health Validator \(WSHV\) Protocol Specification](#)", April 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](#)", July 2006.

[MSDN-OSVERSIONINFOEX] Microsoft Corporation, "OSVERSIONINFOEX" Structure, <http://msdn.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 in which 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 that the health state of the client 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-layer 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 by sending 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 (SHAs)**) 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 (SHVs)**) 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. Windows includes one specific SHA to report the health state of the system and one specific SHV to evaluate the health state. These are the Windows Security Health Agent (WSHA) and Windows Security Health Validator (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 that uses a **Health Policy Server** or to an

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 that the health state of the client 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-layer 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 by sending 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 (SHAs)**) 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 (SHVs)**) 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. Windows includes one specific SHA to report the health state of the system and one specific SHV to evaluate the health state. These are the Windows Security Health Agent (WSHA) and Windows Security Health Validator (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 that uses 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 message. <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 that 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 only 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) by 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 provide 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 subheader, 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 versions.

When a server receives a request message from a client, the server 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).

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 message. <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 that 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 only 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) by 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 provide 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 subheader, 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 versions.

When a server receives a request message from a client, the server 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]

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 messages 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 message 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 set, and MUST be present. Each TLV in an SoHReportEntry/SoHRReportEntry set is called an **SoHAttribute/SoHRAttribute**. There MAY be zero or more SoHAttributes per SoHReportEntry set (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 SoHAttributes per SoHRReportEntry set.

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

A graphic representation of the top-level structure follows.

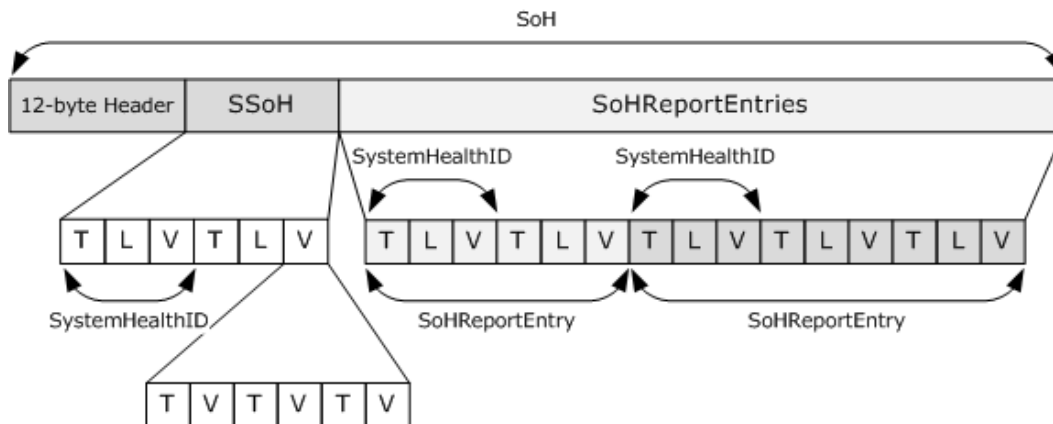


Figure 1: SoH shown without sub-mode header

See section 2.2.7.

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 messages 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 message 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 set, and MUST be present. Each TLV in an SoHReportEntry/SoHRReportEntry set is called an [SoHAttribute/SoHRAttribute](#). There MAY be zero or more SoHAttributes per SoHReportEntry set (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 set.

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

A graphic representation of the top-level structure follows.

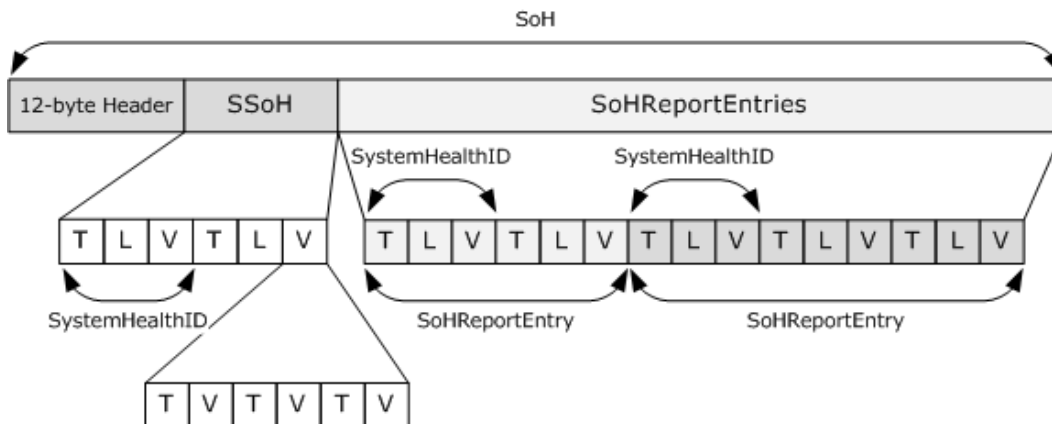


Figure 1: SoH shown without sub-mode header

See section 2.2.7.

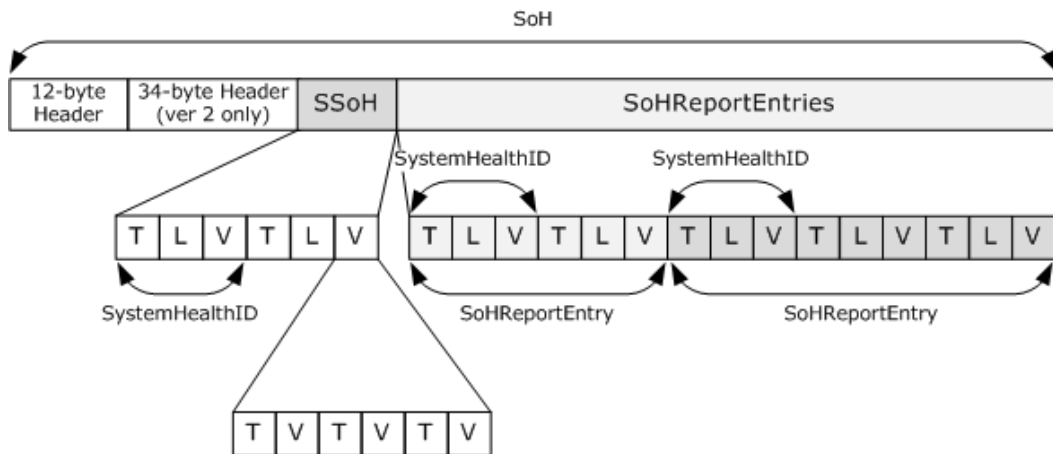
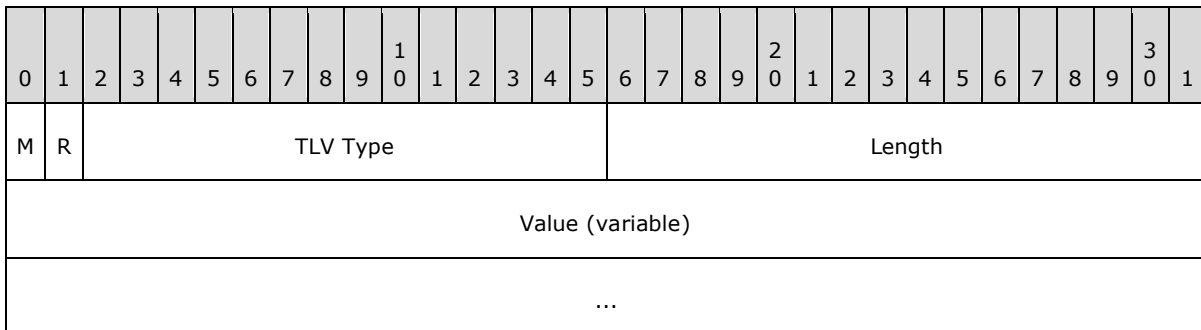


Figure 2: SoH shown with sub-mode header

See section [2.2.7](#).

2.2.1 Type-Length-Value (TLV) Packet

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



M (1 bit): The **M** bit **MUST** be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

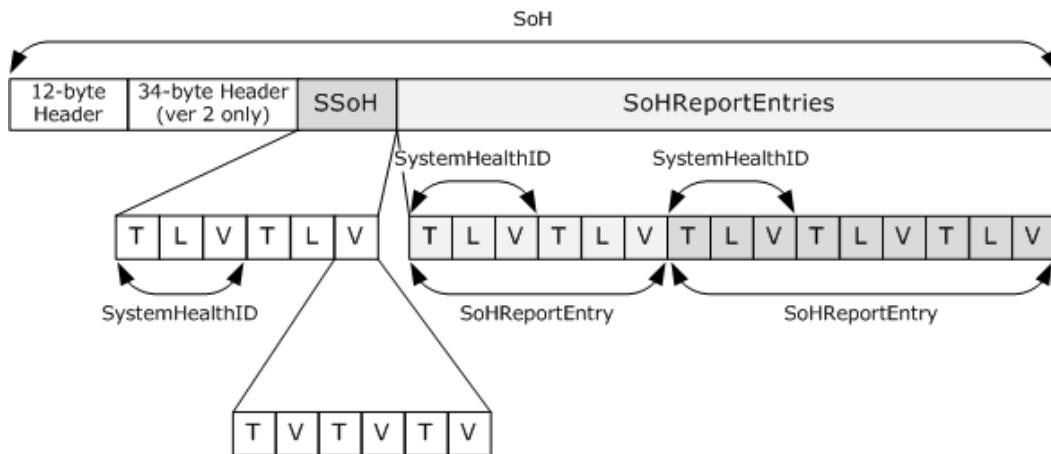
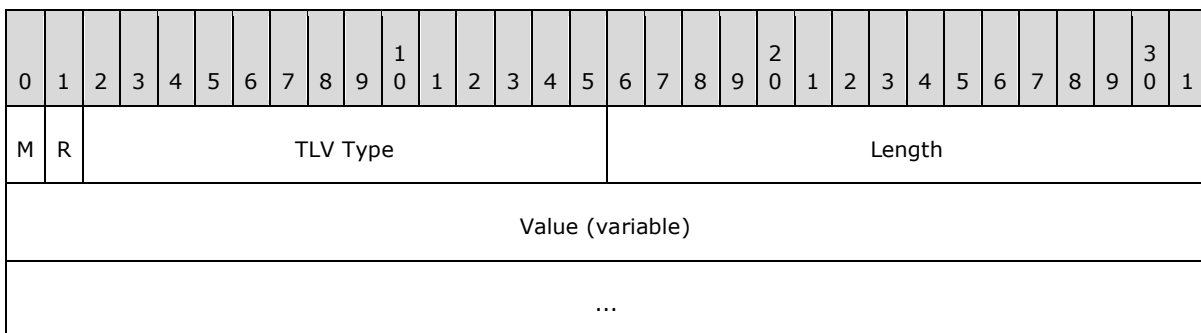


Figure 2: SoH shown with sub-mode header

See section [2.2.7](#).

2.2.1 Type-Length-Value (TLV) Packet

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



M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

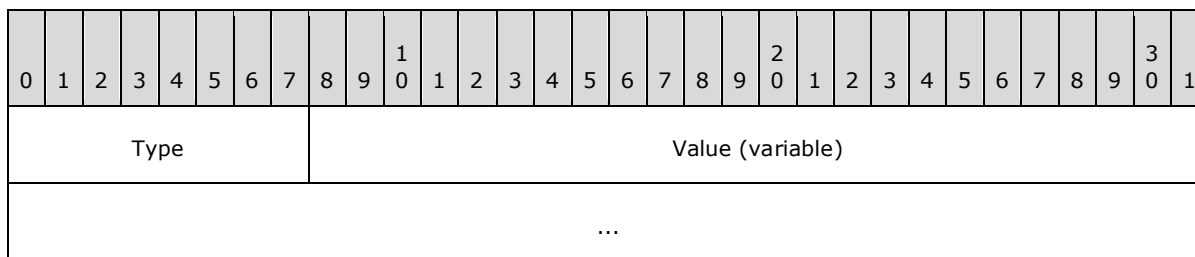
TLV Type (14 bits): A 14-bit unsigned integer that MUST indicate the type of data in the **Value** field. Valid **TLV Type** values MUST be one of those specified in section [2.2.3](#).

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 packet 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](#).



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** field MUST correspond to the type as defined by the **Type** field. 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 through 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 set.
4	Compliance-Result-Codes	Result codes specifying if 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.

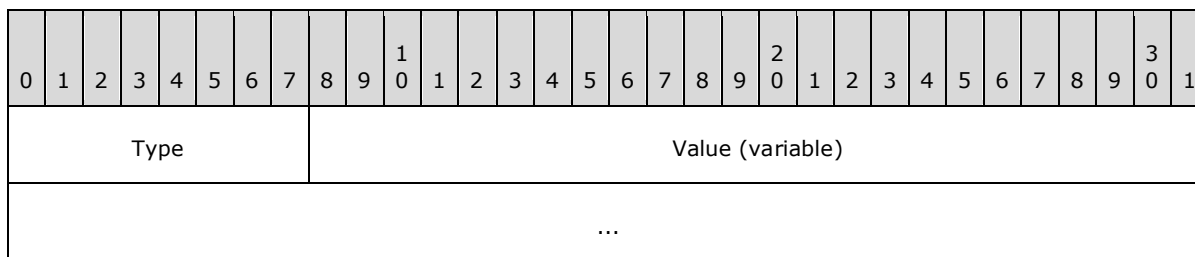
A list of optional TLVs are provided in section [2.2.3.5](#).

2.2.3.1 System-Health-ID Packet

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

2.2.2 Type-Value (TV) Packet

The following packet 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](#).



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** field MUST correspond to the type as defined by the **Type** field. 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 through 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 set.
4	Compliance-Result-Codes	Result codes specifying if 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](#) set (for the SoH or SoHR message) MUST be the first TLV present in the SoHReportEntry/SoHRReportEntry.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value																															

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/SoHRRReportEntry. The value MUST be formatted as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IANA SMI Code for Vendor																								Id							

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 message is used to specify whether the client computer is compliant with policy. An SoHR message MUST contain this attribute, a [Failure Category](#) attribute, or both. An SoH MAY contain this attribute.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															
...																															

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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value																															

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/SoHRRReportEntry. The value MUST be formatted as follows.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IANA SMI Code for Vendor																								Id							

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 message is used to specify whether the client computer is compliant with policy. An SoHR message MUST contain this attribute, a [Failure Category](#) attribute, or both. An SoH MAY contain this attribute.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															
...																															

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.

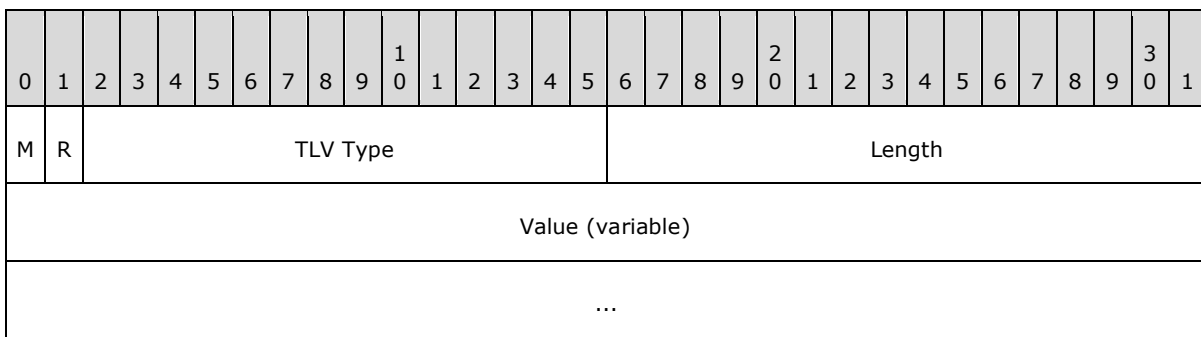
Value (variable): An array of **HRESULT** values that indicate the results of evaluation by the server.

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 can 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 message. It is also present in [SSoH](#) and [SSoHR](#).



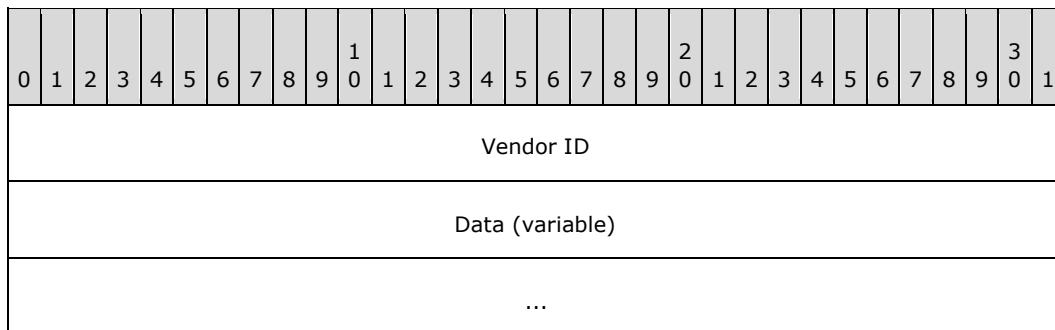
M (1 bit): The **M** bit MUST be set to zero.

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): A 32-bit unsigned integer that SHOULD specify the assigned IANA-assigned SMI for the vendor whose data is to be specified in the **Data** field. [<4>](#) NAP

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.

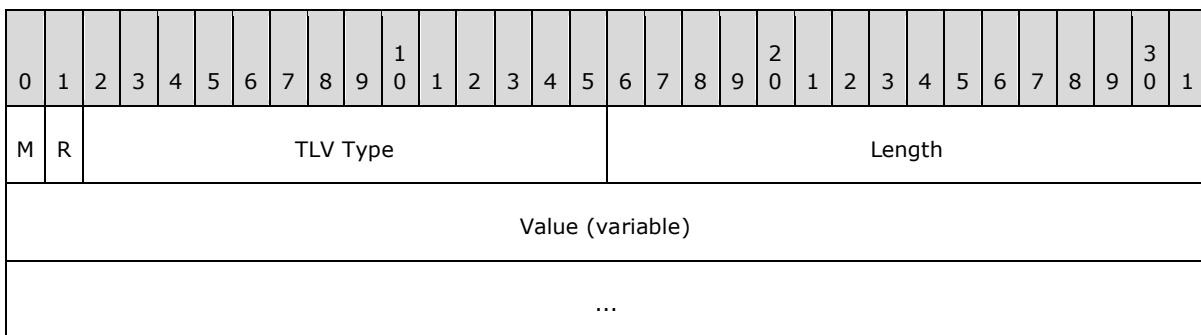
Value (variable): An array of **HRESULT** values that indicate the results of evaluation by the server.

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 can 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 message. It is also present in [SSoH](#) and [SSoHR](#).



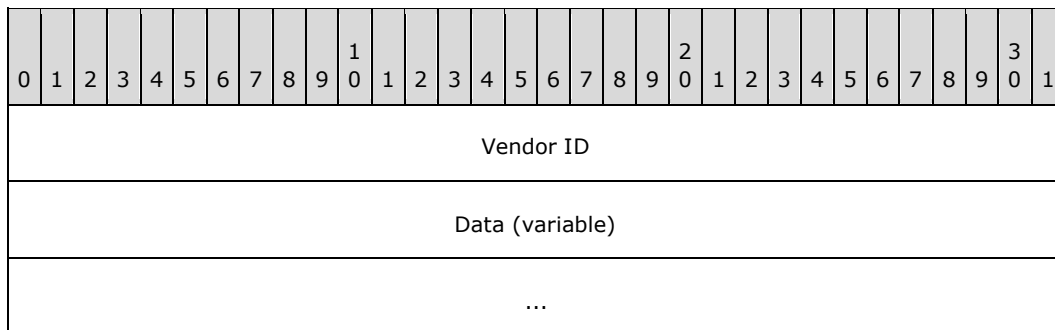
M (1 bit): The **M** bit MUST be set to zero.

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): A 32-bit unsigned integer that SHOULD specify the assigned IANA-assigned SMI for the vendor whose data is to be specified in the **Data** field. <4> NAP

does not interpret this field in SoH or SoHR messages. The vendor MAY use it for any purpose. However, NAP does interpret this field in SSoH and SSoHR.

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 message MUST contain this TLV, a [Compliance-Result-Code](#) TLV, or both. This attribute MAY be present in an SoH message.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value																															

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 components or communications of the client or server
2	Failure due to client component.
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 that follows 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 SoH or SoHR messages, the lengths specified in the following sections for each optional TLV MUST be honored. [<5>](#)

does not interpret this field in SoH or SoHR messages. The vendor MAY use it for any purpose. However, NAP does interpret this field in SSoH and SSoHR.

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 message MUST contain this TLV, a [Compliance-Result-Code](#) TLV, or both. This attribute MAY be present in an SoH message.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value																															

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 components or communications of the client or server
2	Failure due to client component.
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 that follows 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 SoH or SoHR messages, 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 Fix-up 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 computer was last updated which is 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 message 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 SoHRRReportEntry set	Variable
15	IPv6 Fix-up Servers, specified in section 2.2.3.5.12 .	IPv6 addresses of the fix-up servers	Variable
16-255	Reserved	Reserved	N/A

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

2.2.3.5.1 Optional TLV 0: Reserved

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

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 Fix-up 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 computer was last updated which is 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 message 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 SoHRRReportEntry set	Variable
15	IPv6 Fix-up Servers, specified in section 2.2.3.5.12 .	IPv6 addresses of the fix-up servers	Variable
16-255	Reserved	Reserved	N/A

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

2.2.3.5.1 Optional TLV 0: Reserved

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Reserved																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 Optional TLV 1: Reserved packet is reserved for future use and MUST be ignored if received in error.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Reserved																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Reserved																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 Optional TLV 1: Reserved packet is reserved for future use and MUST be ignored if received in error.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Reserved																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

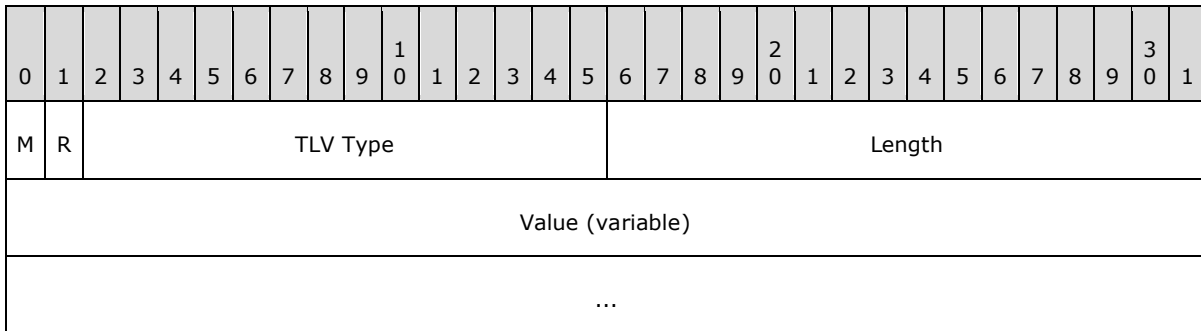
TLV Type (14 bits): Reserved. For this packet type, MUST always be 1.

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 Fix-up Servers

The Optional TLV 3: IPv4 Fix-up Servers packet provides the addresses of the fix-up servers. An SoH message MAY contain 0 plus this attribute. An SoHR message MAY contain 0 or 1 plus this attribute. An SHA can use this information to perform remediation.



M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

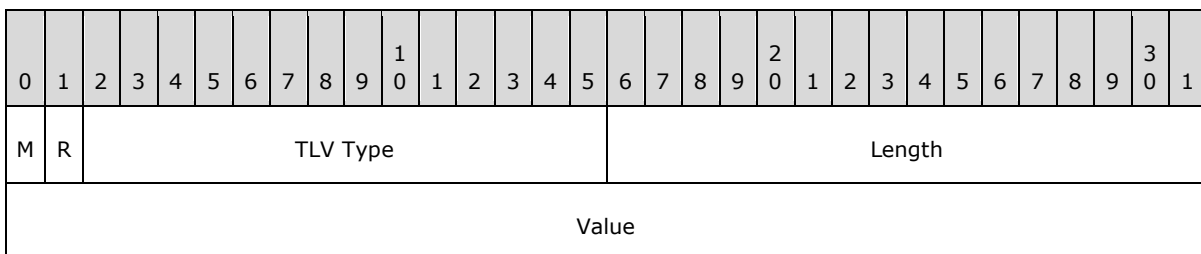
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 3.

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 4-byte IPv4 addresses of the fix-up servers.

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

The Optional TLV 5: Time-of-Last-Update packet specifies the UTC time when the client computer was last updated which is measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC). [<6>](#)

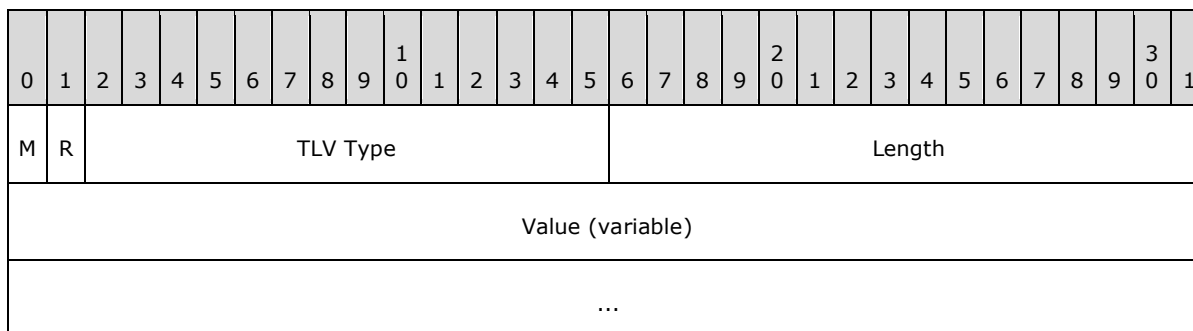


Length (2 bytes): A 16-bit unsigned integer that MUST specify the length, in bytes, of the Reserved field. For this packet type, MUST be set to 4.

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 Fix-up Servers

The Optional TLV 3: IPv4 Fix-up Servers packet provides the addresses of the fix-up servers. An SoH message MAY contain 0 plus this attribute. An SoHR message MAY contain 0 or 1 plus this attribute. An SHA can use this information to perform remediation.



M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

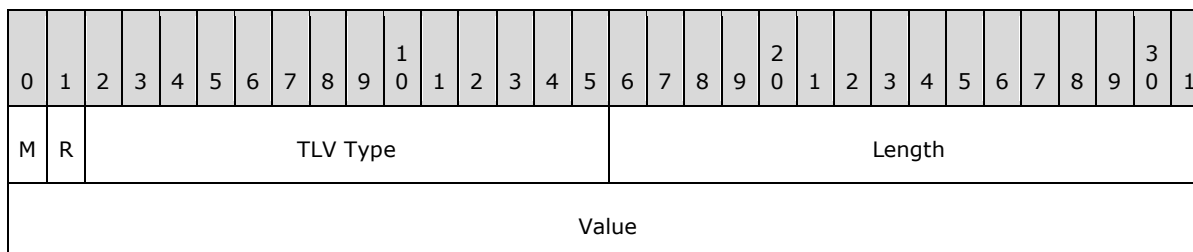
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 3.

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 4-byte IPv4 addresses of the fix-up servers.

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

The Optional TLV 5: Time-of-Last-Update packet specifies the UTC time when the client computer was last updated which is measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC). [<6>](#)



...

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 computer was last updated which is measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC).

2.2.3.5.5 Optional TLV 6: Client-ID

The Optional TLV 6: Client-ID packet specifies the client identifier. <7>

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															
...																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

...

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 computer was last updated which is measured as the number of 100-nanosecond intervals since January 1, 1601 (UTC).

2.2.3.5.5 Optional TLV 6: Client-ID

The Optional TLV 6: Client-ID packet specifies the client identifier. <7>

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															
...																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

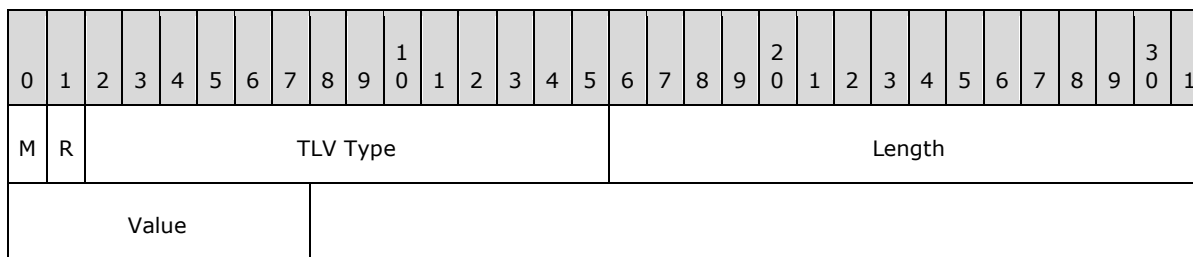
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 6.

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 Optional TLV 8: 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 MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

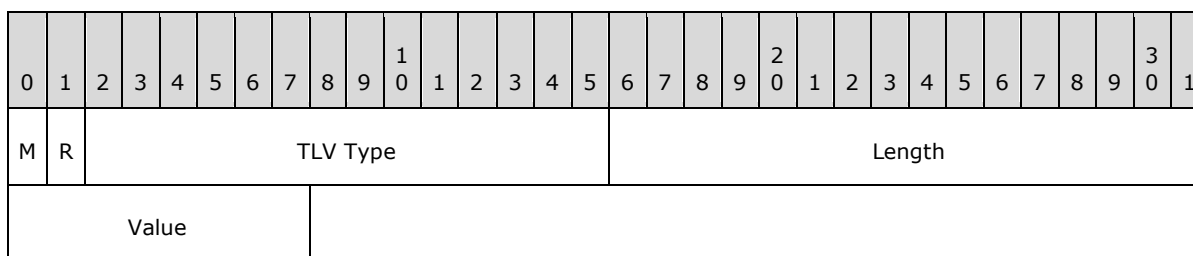
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 8.

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

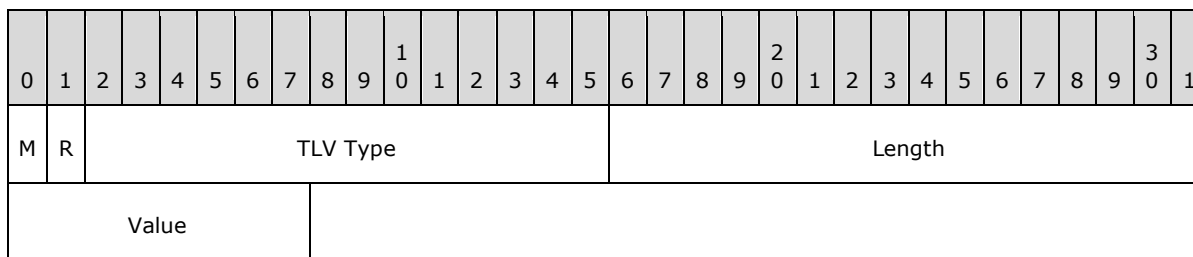


M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory TLV.
1	This is a mandatory TLV.

2.2.3.5.6 Optional TLV 8: Health-Class

The Optional TLV 8: 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 MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

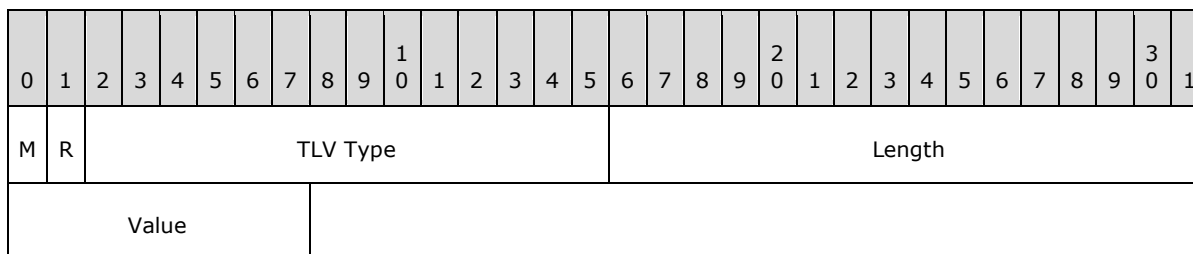
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 8.

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



M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

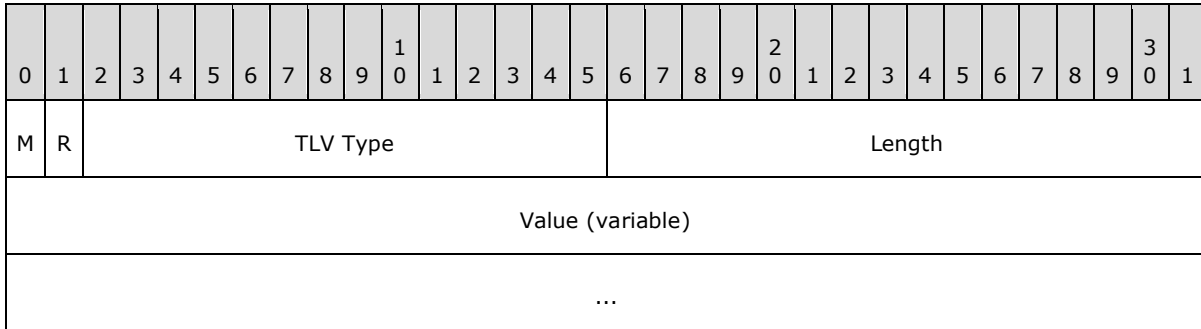
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 9.

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 Optional TLV 10: Product-Name packet specifies the name of the product installed on the client computer. [<10>](#)



M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

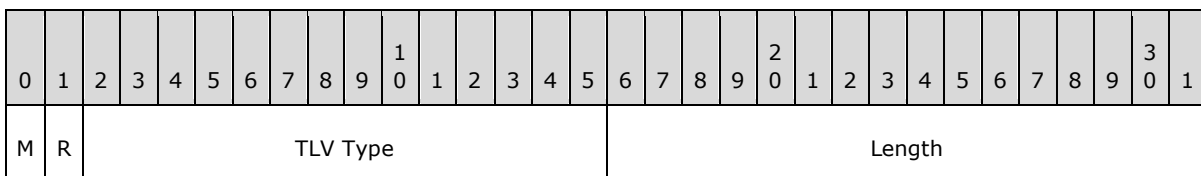
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 10.

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 Optional TLV 11: Health Class Status packet specifies the Status code for the health-class type given by the Health-Class TLV. [<11>](#)



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

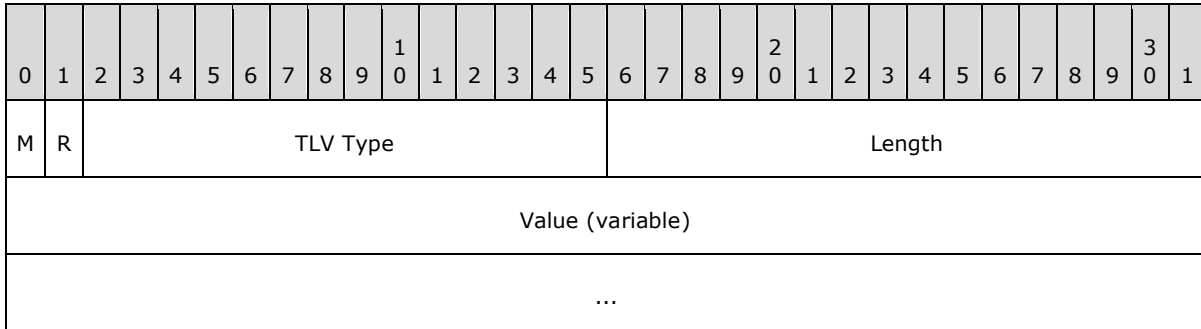
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 9.

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



M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

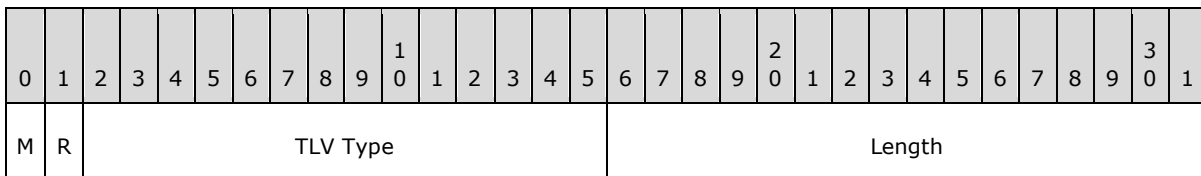
TLV Type (14 bits): The **TLV Type** for this packet type MUST always be 10.

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



Value (variable)
...

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 Optional TLV 12: SOH Generation Time packet specifies the UTC time when the SoH message was generated. [<12>](#)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
M	R	TLV Type														Length																	
Value																																	
...																																	

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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.

Value (variable)
...

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 Optional TLV 12: SOH Generation Time packet specifies the UTC time when the SoH message was generated. [<12>](#)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value																															
...																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 message 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 Optional TLV 13: Error Codes packet returns a set of error codes of type [HRESULT.<13>](#)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															
...																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 HRESULT values.

2.2.3.5.12 Optional TLV 15: IPv6 Fix-up Servers

The Optional TLV 15: IPv6 Fix-up Servers packet specifies the addresses of the IPv6 Fix-up Servers. An SoH message MAY contain 0 plus this attribute. An SoHR message MAY contain 0 or 1 plus this attribute. An SHA can use this information to perform remediation.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															

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 message 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 Optional TLV 13: Error Codes packet returns a set of error codes of type [HRESULT.<13>](#)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															
...																															

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 HRESULT values.

2.2.3.5.12 Optional TLV 15: IPv6 Fix-up Servers

The Optional TLV 15: IPv6 Fix-up Servers packet specifies the addresses of the IPv6 Fix-up Servers. An SoH message MAY contain 0 plus this attribute. An SoHR message MAY contain 0 or 1 plus this attribute. An SHA can use this information to perform remediation.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
M	R	TLV Type														Length															
Value (variable)																															

...

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 SSoHRAAttribute

The SSoHAttribute/SSoHRAAttribute elements are used to construct valid [SSoH](#) and [SSoHR TLVs](#). SSoHAttribute/SSoHRAAttribute 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 which have values from 0 through 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](#) message, and MAY be present in an [SSoHR](#) message [<15>](#).

...

M (1 bit): The **M** bit MUST be set to one of the following values.

Value	Meaning
0	This is a nonmandatory 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 SSoHRAAttribute

The SSoHAttribute/SSoHRAAttribute elements are used to construct valid [SSoH](#) and [SSoHR TLVs](#). SSoHAttribute/SSoHRAAttribute 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 which have values from 0 through 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](#) message, and MAY be present in an [SSoHR](#) message [<15>](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
osVersionMajor																															
osVersionMinor																															
osVersionBuild																															
spVersionMajor																spVersionMinor															
procArch																															

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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
osVersionMajor																															
osVersionMinor																															
osVersionBuild																															
spVersionMajor																spVersionMinor															
procArch																															

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](#) message and the [SSoHR](#) message.

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Reserved1								ExtState				f	qState				ProbTime															
...																																
...																urlLenInBytes																
url (variable)																																
...																																

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

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

Value	Meaning
0	No evaluation was done (this will be used if the network policy server (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
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](#) message and the [SSoHR](#) message.

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Reserved1								ExtState				f	qState				ProbTime															
...																																
...																urlLenInBytes																
url (variable)																																
...																																

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

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

Value	Meaning
0	No evaluation was done (this will be used if the network policy server (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): A 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 terminating 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](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Reserved			r	vers																															

Reserved (3 bits): The three bits of the **Reserved** field are reserved, MUST be set to zero when sending and ignored upon receipt.

r (1 bit): A 1-bit response/request flag. The value indicates if the attribute contains a request or a response message. 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 that is set in the header.

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

qState (3 bits): A 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 terminating 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](#).

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved			r	vers																											

Reserved (3 bits): The three bits of the **Reserved** field are reserved, MUST be set to zero when sending and ignored upon receipt.

r (1 bit): A 1-bit response/request flag. The value indicates if the attribute contains a request or a response message. 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 that is 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>](#)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																idList (variable)															
...																															

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 SoHs or 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 Subpacket

The MS-SystemGenerated-Ids subpacket for the MS-SystemGenerated-Ids Packet idList field:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IANA SMI Code for Vendor																								Id							

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 value of the **IANA SMI Code for Vendor** field, allows the routing of an [SoHReportEntry](#) set from a client component to the corresponding server-side component that can deal with it. Similarly, the [SoHRReportEntry](#) set is routed to the originator based on this ID. The value of 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 ID fully identifies the source and destination of each SoHReportEntry and SoHRReportEntry set.

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](#).

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>](#)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																idList (variable)															
...																															

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 SoHs or 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 Subpacket

The MS-SystemGenerated-Ids subpacket for the MS-SystemGenerated-Ids Packet idList field:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IANA SMI Code for Vendor																								Id							

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 value of the **IANA SMI Code for Vendor** field, allows the routing of an [SoHReportEntry](#) set from a client component to the corresponding server-side component that can deal with it. Similarly, the [SoHRReportEntry](#) set is routed to the originator based on this ID. The value of 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 ID fully identifies the source and destination of each SoHReportEntry and SoHRReportEntry set.

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](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																machineName (variable)															
...																															

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](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
correlationId																															
...																															
...																															
...																															
...																															
...																															

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 packet is a list of identifiers of services that can evaluate SoH messages 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](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																machineName (variable)															
...																															

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](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
correlationId																															
...																															
...																															
...																															
...																															
...																															

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 packet is a list of identifiers of services that can evaluate SoH messages 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](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																idList (variable)															
...																															

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 Subpacket

The MS-Installed-Shvs subpacket provides the information about the identifiers contained in the **idList** field of the [MS-Installed-Shvs](#) packet.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IANA SMI Code for Vendor																								Id							

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>

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Reserved																															
ProductType																															

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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Length																idList (variable)															
...																															

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 Subpacket

The MS-Installed-Shvs subpacket provides the information about the identifiers contained in the **idList** field of the [MS-Installed-Shvs](#) packet.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IANA SMI Code for Vendor																								Id							

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>

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Reserved																															
ProductType																															

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 whose value field is the rest of the content for the message.

2.2.5.1 SoH Header

This is the SoH Header packet for the SoH message.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Rvd		Outer Type														Length															
IANA SMI Code																															
Inner Type																Inner Length															
Value (variable)																															
...																															

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 value of the **Inner Type** field, 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 the 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

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 whose value field is the rest of the content for the message.

2.2.5.1 SoH Header

This is the SoH Header packet for the SoH message.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Rvd		Outer Type														Length															
IANA SMI Code																															
Inner Type																Inner Length															
Value (variable)																															
...																															

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 value of the **Inner Type** field, 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 the 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

Value	Meaning
	SoHReportEntry (0 plus)
0x0002	SoH Mode Subheader SSoH SoHReportEntry (0 plus)

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 plus)

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

- SoH Mode Subheader
- SSoH
- SoHReportEntry (0 plus)

2.2.5.2 SoHReportEntry

The SoHReportEntry message is used to represent a set of [SoHAttributes](#); it has no header of its own and is simply constructed as a set of SoHAttributes.

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

The SoHReportEntry message MUST contain one or more additional SoHAttributes as follows (see section [2.2](#)):

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

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 whose value field is the rest of the content for the message.

2.2.6.1 SoHR Header

This is the SoHR Header for the SoHR packet.

Value	Meaning
	SoHReportEntry (0 plus)
0x0002	SoH Mode Subheader SSoH SoHReportEntry (0 plus)

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 plus)

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

- SoH Mode Subheader
- SSoH
- SoHReportEntry (0 plus)

2.2.5.2 SoHReportEntry

The SoHReportEntry message is used to represent a set of [SoHAttributes](#); it has no header of its own and is simply constructed as a set of SoHAttributes.

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

The SoHReportEntry message MUST contain one or more additional SoHAttributes as follows (see section [2.2](#)):

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

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 whose value field is the rest of the content for the message.

2.2.6.1 SoHR Header

This is the SoHR Header for the SoHR packet.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Rvd		Outer Type														Length															
IANA SMI Code																															
Inner Type																Inner Length															
Value (variable)																															
...																															

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 value of the **Inner Type** field, 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 the 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 SoHRRReportEntry (0 plus)
0x0002	SoH Mode Subheader SSoHR SoHRRReportEntry (0 plus)

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
- SoHRRReportEntry (0 plus)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Rvd		Outer Type														Length															
IANA SMI Code																															
Inner Type																Inner Length															
Value (variable)																															
...																															

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 value of the **Inner Type** field, 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 the 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 SoHRRReportEntry (0 plus)
0x0002	SoH Mode Subheader SSoHR SoHRRReportEntry (0 plus)

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
- SoHRRReportEntry (0 plus)

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

- SoH Mode Subheader
- SSoHR
- SoHRReportEntry (0 plus)

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](#) packet MUST be the first SoHR attribute.

The System-Health-Id MUST be followed by either a [Compliance-Result-Codes](#) packet, a [Failure Category](#) packet, or both.

The SoHRReportEntry message MUST contain two or more additional SoHR attributes as follows (see section [2.2](#)):

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

2.2.7 SoH Mode Subheader

The SoH Mode Subheader is used to represent information about the information following it.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Rvd	Outer Type															Length															
IANA SMI Code																															
Value																															
...																															
...																															
...																															
...																															
...																															
...																															

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

- SoH Mode Subheader
- SSoHR
- SoHRReportEntry (0 plus)

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](#) packet MUST be the first SoHR attribute.

The System-Health-Id MUST be followed by either a [Compliance-Result-Codes](#) packet, a [Failure Category](#) packet, or both.

The SoHRReportEntry message MUST contain two or more additional SoHR attributes as follows (see section [2.2](#)):

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

2.2.7 SoH Mode Subheader

The SoH Mode Subheader is used to represent information about the information following it.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Rvd	Outer Type															Length															
IANA SMI Code																															
Value																															
...																															
...																															
...																															
...																															
...																															
...																															

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 distributed as follows:

~~Correlation ID:~~ 24 bytes that MUST have the same value as the **Value** field of the [MS-CorrelationId](#) message.

~~Intent Flag:~~ 1 byte that MUST be 0x01 for SoH request messages and 0x00 for SoHR response messages.

Value	Meaning
0x01	SoH request message
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 [SoHAttributes](#) and [SSoHAttributes](#):

- **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 SSoHAttributes:
 - **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-MachineName** (see section 2.2.4.5).
 - **MS-CorrelationId** (see section 2.2.4.6).

The **Value** field MAY contain the following SSoHAttributes:

- **MS-SystemGenerated-Ids** (optional; see section 2.2.4.4).

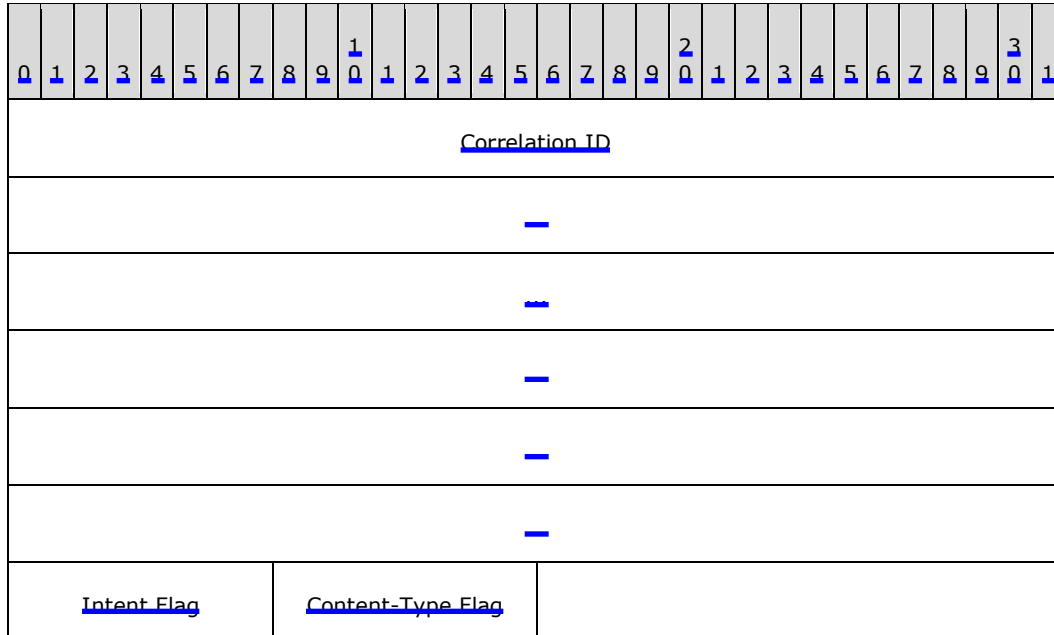
Rvd (2 bits): The **Rvd** field is reserved, and MUST be set to zero and ignored on receipt.

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 distributed as follows:



Correlation ID (24 bytes): 24 bytes that MUST have the same value as the **Value** field of the [MS-CorrelationId](#) message.

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

Value	Meaning
0x01	SoH request message
0x00	SoHR response message

Content-Type Flag (1 byte): 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.

- **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](#)).

The **Value** field SHOULD contain the following SSoHR attributes:

- **MS-Installed-Shvs** (optional; see section [2.2.4.7](#)).<24>

The SSoH message has no header of its own and is simply constructed as the following ordered sequence of [SoHAttributes](#) and [SSoHAttributes](#):

- **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 SSoHAttributes:
 - **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-MachineName** (see section [2.2.4.5](#)).
 - **MS-CorrelationId** (see section [2.2.4.6](#)).

The **Value** field MAY contain the following SSoHAttributes:

 - **MS-SystemGenerated-Ids** (optional; see section [2.2.4.4](#)).
 - **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](#)).

The **Value** field SHOULD contain the following SSoHR attributes:

 - **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 message 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

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

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 message corresponds to an SoH message that was sent.

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

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 message 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

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

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 message corresponds to an SoH message that was sent.

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

3.2.2 Timers

For more information about timers, see section [3.1.2](#).

3.2.3 Initialization

For more information about initialization, see section [3.1.3](#).

3.2.4 Higher-Layer Triggered Events

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

- A user reboots a machine.
- The status of the client changes. For example, the firewall on the client is turned off.

In addition, events specific to the transport mechanism 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 in accordance with locally configured policy by 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.

3.2.2 Timers

For more information about timers, see section [3.1.2](#).

3.2.3 Initialization

For more information about initialization, see section [3.1.3](#).

3.2.4 Higher-Layer Triggered Events

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

- A user reboots a machine.
- The status of the client changes. For example, the firewall on the client is turned off.

In addition, events specific to the transport mechanism 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 in accordance with locally configured policy by 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](#) attribute or a [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

None.

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

None.

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](#) attribute or a [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

None.

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

None.

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 the health of the client can be evaluated.

A health policy server MUST do the following:

- Receive and process SoHs. [<30>](#)
- Create SoHRs. [<31>](#)
- 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](#) that it sends is properly formed, including validating the length of each attribute. The SoH server MUST include at least a valid [Compliance-Result-Codes](#) attribute or a [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

None.

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.

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 the health of the client can be evaluated.

A health policy server MUST do the following:

- Receive and process SoHs. [<30>](#)
- Create SoHRs. [<31>](#)
- 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](#) that it sends is properly formed, including validating the length of each attribute. The SoH server MUST include at least a valid [Compliance-Result-Codes](#) attribute or a [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

None.

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 in the following.

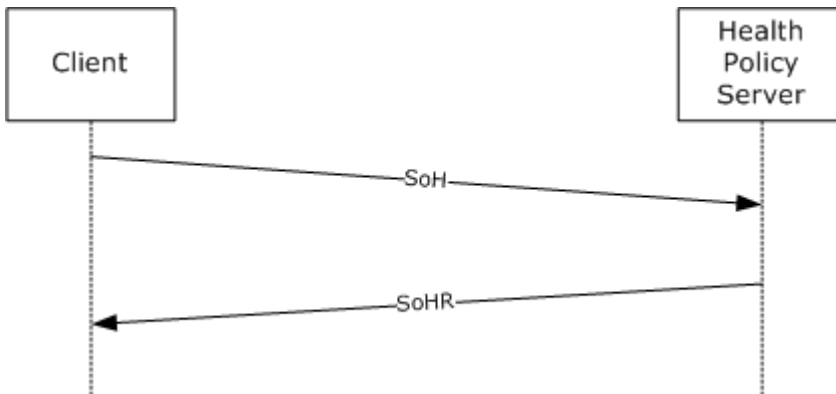


Figure 3: Client SoH request and health policy server response

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 its 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 (that is, the SoHR) to the client by means of the original receiver of the SoH.

Generally, the receipt of an SoHR by the client, 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. After the client is restored to good health, the client can initiate the protocol again.

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 in the following.

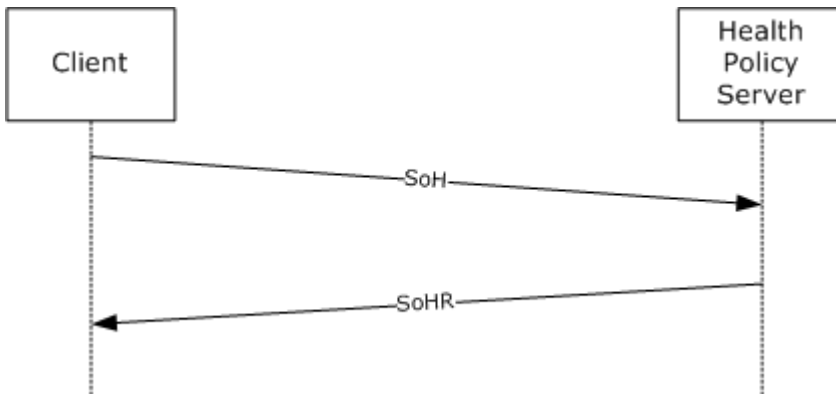


Figure 3: Client SoH request and health policy server response

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 its 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 (that is, the SoHR) to the client by means of the original receiver of the SoH.

Generally, the receipt of an SoHR by the client, 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. After the client is restored to good health, the client can initiate the protocol again.

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 can 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 version 2 for future compatibility and security enhancements.

Implementers can use the protocol in applications where the messages are carried in a transport protocol that does not provide security (for example, DHCP). In such cases, it is important for the implementation 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 protocol provides security for health messages:

- **Passive Observation of Confidential Information.** 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 that 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 attack (MITM)** to successfully view the contents of health messages as they are transmitted.
- **Spoofing.** A 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 the health message. 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 NAS, **EAP server**, or a client.
- **Active Tampering.** For the risks discussed previously, it should not be possible for an attacker to modify the SoHR undetected. Similarly, tampering of the SoH can cause a client to be granted access when it is impermissible, or cause a client to be denied access when it is permitted. The security provided by the transport mechanism should prevent tampering with these messages.
- **Replaying.** A message that causes a client to be granted access can potentially be retransmitted by another client to incorrectly give the client 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 implementers aware of them.

There are no reliable measures that can prevent a denial of service attack on either a client or an NAS. Such attacks can 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).

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 can 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 version 2 for future compatibility and security enhancements.

Implementers can use the protocol in applications where the messages are carried in a transport protocol that does not provide security (for example, DHCP). In such cases, it is important for the implementation 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 protocol provides security for health messages:

- **Passive Observation of Confidential Information.** 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 that 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 attack (MITM)** to successfully view the contents of health messages as they are transmitted.
- **Spoofing.** A 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 the health message. 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 NAS, **EAP server**, or a client.
- **Active Tampering.** For the risks discussed previously, it should not be possible for an attacker to modify the SoHR undetected. Similarly, tampering of the SoH can cause a client to be granted access when it is impermissible, or cause a client to be denied access when it is permitted. The security provided by the transport mechanism should prevent tampering with these messages.
- **Replaying.** A message that causes a client to be granted access can potentially be retransmitted by another client to incorrectly give the client 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 implementers 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 can 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, without broad deployment of trusted hardware, currently there are no effective solutions for this.

5.2 Index of Security Parameters

None.

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, without broad deployment of trusted hardware, currently there are no effective solutions for this.

5.2 Index of Security Parameters

None.

6 Appendix A: Windows Behavior

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

- Windows Vista
- Windows Server 2008
- Windows 7

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 NAP: The Windows implementation of the SoH Client sends version 2 messages by default, but 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 NAP: The client has to be configured to recognize the access methods that are required to enable the NAP solution.

[<3> Section 2.2.3.1](#): The 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, and 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 who specifies the data in the Data field. For example, in the Windows implementation, Windows Security Health Agent (WSHA) uses the following vendor ID format: IANA-assigned SMI vendor code plus 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 or SoHR messages. The Windows implementation does not check the **Value** fields of the optional TLVs in third-party implementations. However, the 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 or SoHR message, the Windows implementation requires that the length of its **Value** field be 8, but does not require specific contents for the **Value** field.

[<6> Section 2.2.3.5.4](#): Third-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs are used in the SoH/SoHR messages. The detailed implementation by MSSHA/SHV can be found in [WSHA SoH](#) and [WSHV SoHR \[MS-WSH\]](#).

6 Appendix A: Windows Behavior

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

- Windows Vista
- Windows Server 2008
- Windows 7

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 NAP: The Windows implementation of the SoH Client sends version 2 messages by default, but 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 NAP: The client has to be configured to recognize the access methods that are required to enable the NAP solution.

[<3> Section 2.2.3.1](#): The 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, and 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 who specifies the data in the Data field. For example, in the Windows implementation, Windows Security Health Agent (WSHA) uses the following vendor ID format: IANA-assigned SMI vendor code plus 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 or SoHR messages. The Windows implementation does not check the **Value** fields of the optional TLVs in third-party implementations. However, the 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 or SoHR message, the Windows implementation requires that the length of its **Value** field be 8, but does not require specific contents for the **Value** field.

[<6> Section 2.2.3.5.4](#): Third-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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](#): The [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 require nor read this TLV when the [SSoHR](#) is parsed.

<16> [Section 2.2.4.2](#): Client resubmission at end-of-probation time. 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 certificate expires at the end of probation time and the client 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](#): The [MS-SystemGenerated-Ids](#) attribute for internal error indication: The Windows client includes the [MS-SystemGenerated-Ids](#) attribute when an internal error occurs while the client is 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 it is 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 Windows Server 2008 and Windows Server 7. The health policy server in Windows includes an API that enables plug-ins that perform an 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.

<9> [Section 2.2.3.5.7](#): Third-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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-party SHA/SHV implementations and MSSHA/SHV may use 0 plus 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 TLVs 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](#): The [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 require nor read this TLV when the [SSoHR](#) is parsed.

<16> [Section 2.2.4.2](#): Client resubmission at end-of-probation time. 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 certificate expires at the end of probation time and the client 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](#): The [MS-SystemGenerated-Ids](#) attribute for internal error indication: The Windows client includes the [MS-SystemGenerated-Ids](#) attribute when an internal error occurs while the client is 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 it is 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 Windows Server 2008 and Windows Server 7. The health policy server in Windows includes an API that enables plug-ins that perform an 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 require 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 that the [SSoH](#) is generated. The Windows implementation 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 and Windows Server 7. The Windows implementation of the health policy server includes [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 API to allow plug-ins to report client state. The Windows SoH client, also called the NAP agent, includes an API that allows plug-ins that report client state to register with the system. These plug-ins are called SHAs. Examples of SHAs include antivirus clients and security update clients. One specific SHA, WSHA, as specified in [\[MS-WSH\]](#), is included in Windows. Each SHA produces an [SoHReportEntry](#) for the state that 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 quarantine enforcement client 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 API that allows plug-ins that transport SoHs to register with the system. 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 that they transport, the QECs can then pass the SoHRs to the NAP agent for processing.

<28> [Section 3.2.5](#): When the NAP agent receives the SoHR, it validates the message 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 [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 NPS. This is implemented in Windows Server 2008 and Windows Server 7. NPS is the Windows RADIUS server. It includes an API 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, 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](#). The health policy server then waits for the SHVs to complete their evaluation of the [SoHReportEntry](#) values.

<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 require 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 that the [SSoH](#) is generated. The Windows implementation 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 and Windows Server 7. The Windows implementation of the health policy server includes [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 API to allow plug-ins to report client state. The Windows SoH client, also called the NAP agent, includes an API that allows plug-ins that report client state to register with the system. These plug-ins are called SHAs. Examples of SHAs include antivirus clients and security update clients. One specific SHA, WSHA, as specified in [\[MS-WSH\]](#), is included in Windows. Each SHA produces an [SoHReportEntry](#) for the state that 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 quarantine enforcement client 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 API that allows plug-ins that transport SoHs to register with the system. 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 that they transport, the QECs can then pass the SoHRs to the NAP agent for processing.

<28> [Section 3.2.5](#): When the NAP agent receives the SoHR, it validates the message 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 [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 NPS. This is implemented in Windows Server 2008 and Windows Server 7. NPS is the Windows RADIUS server. It includes an API 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, 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](#). The health policy server 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 and Windows Server 7. 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 by taking the [Compliance-Result-Codes](#) and [Failure Category](#) attributes as input. The health policy server forms a valid SoHR by 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 message to the NPS to be processed against policy, and sends the message via RADIUS to the SoH client. This is implemented in Windows Server 2008 and Windows Server 7. The health policy server delivers the SoHR to the 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 mechanism 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 and Windows Server 7. The process that Windows Server 2008 uses to evaluate the SoH is specified in section [3.2](#).

<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 and Windows Server 7. 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 by taking the [Compliance-Result-Codes](#) and [Failure Category](#) attributes as input. The health policy server forms a valid SoHR by 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 message to the NPS to be processed against policy, and sends the message via RADIUS to the SoH client. This is implemented in Windows Server 2008 and Windows Server 7. The health policy server delivers the SoHR to the 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 mechanism 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 and Windows Server 7. The process that Windows Server 2008 uses to evaluate the SoH is specified in section [3.2](#).

~~7 Change Tracking~~

~~No table of changes is available. The document is either new or has had no changes since its last release.~~

7 Change Tracking

This section identifies changes made to [MS-SOH] protocol documentation between January 2009 and February 2009 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

Major and minor changes can be described further using the following revision types:

- New content added.
- Content update.
- Content removed.
- New Windows behavior note added.
- Windows behavior note updated.
- Windows behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.

8 Index

A

Abstract data model
 client ([section 3.1.1](#), [section 3.2.1](#))
 server ([section 3.1.1](#), [section 3.3.1](#))
[Applicability statement](#)

C

[Capability negotiation](#)
[Change tracking](#)
 Client
 abstract data model ([section 3.1.1](#), [section 3.2.1](#))
 higher-layer triggered events ([section 3.1.4](#),
[section 3.2.4](#))
 initialization ([section 3.1.3](#), [section 3.2.3](#))
[local events](#)
 message processing ([section 3.1.5](#), [section 3.2.5](#))
 overview ([section 3.1](#), [section 3.2](#))
 sequencing rules ([section 3.1.5](#), [section 3.2.5](#))
 timer events ([section 3.1.6](#), [section 3.2.6](#))
 timers ([section 3.1.2](#), [section 3.2.2](#))
[Client ID packet](#)
[Compliance Result Codes packet](#)

D

Data model - abstract
 client ([section 3.1.1](#), [section 3.2.1](#))
 server ([section 3.1.1](#), [section 3.3.1](#))

E

[Error Codes packet](#)
[Examples](#)

F

[Failure Category packet](#)
[Fields - vendor-extensible](#)

G

[Glossary](#)

H

Header
[SoH](#)
[SoHR](#)
[sub - SoH mode](#)
[Health Class packet](#)
[Health Class Status packet](#)
 Higher-layer triggered events
 client ([section 3.1.4](#), [section 3.2.4](#))
 server ([section 3.1.4](#), [section 3.3.4](#))

I

[Implementers - security considerations](#)
[Informative references](#)

Initialization
 client ([section 3.1.3](#), [section 3.2.3](#))
 server ([section 3.1.3](#), [section 3.3.3](#))

[Introduction](#)
[IPv4 Fixup Servers packet](#)
[IPV6 Fix up Servers packet](#)

L

Local events
[client](#)
[server](#)

M

Message processing
 client ([section 3.1.5](#), [section 3.2.5](#))
 server ([section 3.1.5](#), [section 3.3.5](#))

Messages
[overview](#)
[syntax](#)
[transport](#)
[MS CorrelationId packet](#)
[MS Installed Shvs packet](#)
[MS Installed Shvs Sub packet](#)
[MS Machine Inventory packet](#)
[MS Machine Inventory Ex packet](#)
[MS MachineName packet](#)
[MS Packet Info packet](#)
[MS Quarantine State packet](#)
[MS SystemGenerated Ids packet](#)
[MS SystemGenerated Ids Sub packet](#)

N

[Normative references](#)

O

[Optional TLVs](#)
[Overview \(synopsis\)](#)

P

[Parameters - security](#)
[Preconditions](#)
[Prerequisites](#)
[Product Name packet](#)

R

Receiving
 SoHRs ([section 3.2.5.4](#), [section 3.3.5.4](#))
 SoHs ([section 3.2.5.2](#), [section 3.3.5.2](#))

References
[informative](#)
~~[normative](#)~~

~~Content removed for template compliance.~~

~~Obsolete document removed.~~

~~Editorial changes always have the revision type "Editorially updated."~~

~~Some important terms used in revision type descriptions are defined as follows:~~

~~**Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.~~

~~**Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.~~

~~Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.~~

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.7 SoH Mode Subheader	Changed the presentation of the Value field to use a diagram to show the subfields Correlation ID, Intent Flag, and Content-Type Flag.	N	Editorially updated.

[overview](#)[Relationship to other protocols](#)**S**[Security](#)

Sending

SoHRs ([section 3.2.5.3](#), [section 3.3.5.3](#))SoHs ([section 3.2.5.1](#), [section 3.3.5.1](#))

Sequencing rules

client ([section 3.1.5](#), [section 3.2.5](#))server ([section 3.1.5](#), [section 3.3.5](#))

Server

abstract data model ([section 3.1.1](#), [section 3.3.1](#))higher-layer triggered events ([section 3.1.4](#),
[section 3.3.4](#))initialization ([section 3.1.3](#), [section 3.3.3](#))[local events](#)message processing ([section 3.1.5](#), [section 3.3.5](#))overview ([section 3.1](#), [section 3.3](#))sequencing rules ([section 3.1.5](#), [section 3.3.5](#))timer events ([section 3.1.6](#), [section 3.3.6](#))timers ([section 3.1.2](#), [section 3.3.2](#))[Software Version packet](#)[SoH](#)[SOH Generation Time packet](#)[SoH Header packet](#)[SoH Mode Sub Header packet](#)[SoHAttributes](#)[SoHR](#)[SoHR Header packet](#)[SoHRAttributes](#)[SoHRReportEntry](#)[SoHRReportEntry](#)[SSoH](#)[SSoHAttribute](#)[SSoHR](#)[SSoHRAttribute](#)[Standards assignments](#)[Subheader - SoH mode](#)[Syntax - message](#)[System Health ID packet](#)**T**[Time of Last Update packet](#)

Timer events

client ([section 3.1.6](#), [section 3.2.6](#))server ([section 3.1.6](#), [section 3.3.6](#))

Timers

client ([section 3.1.2](#), [section 3.2.2](#))server ([section 3.1.2](#), [section 3.3.2](#))[TLV packet](#)[TLV0 Reserved packet](#)[TLV1 Reserved packet](#)[TLVs](#)[Tracking changes](#)[Transport - message](#)

Triggered events - higher layer

client ([section 3.1.4](#), [section 3.2.4](#))server ([section 3.1.4](#), [section 3.3.4](#))[TV packet](#)**V**[Vendor Specific packet](#)[Vendor-extensible fields](#)[Versioning](#)**W**[Windows behavior](#)

8 Index

A

Abstract data model
 client ([section 3.1.1](#), [section 3.2.1](#))
 server ([section 3.1.1](#), [section 3.3.1](#))
[Applicability statement](#)

C

[Capability negotiation](#)
[Change tracking](#)
 Client
 abstract data model ([section 3.1.1](#), [section 3.2.1](#))
 higher-layer triggered events ([section 3.1.4](#),
[section 3.2.4](#))
 initialization ([section 3.1.3](#), [section 3.2.3](#))
[local events](#)
 message processing ([section 3.1.5](#), [section 3.2.5](#))
 overview ([section 3.1](#), [section 3.2](#))
 sequencing rules ([section 3.1.5](#), [section 3.2.5](#))
 timer events ([section 3.1.6](#), [section 3.2.6](#))
 timers ([section 3.1.2](#), [section 3.2.2](#))
[Client ID packet](#)
[Compliance Result Codes packet](#)

D

Data model - abstract
 client ([section 3.1.1](#), [section 3.2.1](#))
 server ([section 3.1.1](#), [section 3.3.1](#))

E

[Error Codes packet](#)
[Examples](#)

F

[Failure Category packet](#)
[Fields - vendor-extensible](#)

G

[Glossary](#)

H

Header
[SoH](#)
[SoHR](#)
[sub - SoH mode](#)
[Health Class packet](#)
[Health Class Status packet](#)
 Higher-layer triggered events
 client ([section 3.1.4](#), [section 3.2.4](#))
 server ([section 3.1.4](#), [section 3.3.4](#))

I

[Implementers - security considerations](#)
[Informative references](#)
 Initialization
 client ([section 3.1.3](#), [section 3.2.3](#))
 server ([section 3.1.3](#), [section 3.3.3](#))
[Introduction](#)
[IPv4 Fixup Servers packet](#)
[IPV6 Fix up Servers packet](#)

L

Local events
[client](#)
[server](#)

M

Message processing
 client ([section 3.1.5](#), [section 3.2.5](#))
 server ([section 3.1.5](#), [section 3.3.5](#))
 Messages
[overview](#)
[syntax](#)
[transport](#)
[MS CorrelationId packet](#)
[MS Installed Shvs packet](#)
[MS Installed Shvs Sub packet](#)
[MS Machine Inventory packet](#)
[MS Machine Inventory Ex packet](#)
[MS MachineName packet](#)
[MS Packet Info packet](#)
[MS Quarantine State packet](#)
[MS SystemGenerated Ids packet](#)
[MS SystemGenerated Ids Sub packet](#)

N

[Normative references](#)

O

[Optional TLVs](#)
[Overview \(synopsis\)](#)

P

[Parameters - security](#)
[Preconditions](#)
[Prerequisites](#)
[Product Name packet](#)

R

Receiving
 SoHRs ([section 3.2.5.4](#), [section 3.3.5.4](#))
 SoHs ([section 3.2.5.2](#), [section 3.3.5.2](#))
 References
[informative](#)

[normative
overview](#)

[Relationship to other protocols](#)

S

[Security](#)

Sending

SoHRs ([section 3.2.5.3](#), [section 3.3.5.3](#))

SoHs ([section 3.2.5.1](#), [section 3.3.5.1](#))

Sequencing rules

client ([section 3.1.5](#), [section 3.2.5](#))

server ([section 3.1.5](#), [section 3.3.5](#))

Server

abstract data model ([section 3.1.1](#), [section 3.3.1](#))

higher-layer triggered events ([section 3.1.4](#),
[section 3.3.4](#))

initialization ([section 3.1.3](#), [section 3.3.3](#))

[local events](#)

message processing ([section 3.1.5](#), [section 3.3.5](#))

overview ([section 3.1](#), [section 3.3](#))

sequencing rules ([section 3.1.5](#), [section 3.3.5](#))

timer events ([section 3.1.6](#), [section 3.3.6](#))

timers ([section 3.1.2](#), [section 3.3.2](#))

[Software_Version packet](#)

[SoH](#)

[SOH_Generation_Time packet](#)

[SoH_Header packet](#)

[SoH_Mode_Sub_Header packet](#)

[SoHAttributes](#)

[SoHR](#)

[SoHR_Header packet](#)

[SoHRAttributes](#)

[SoHReportEntry](#)

[SoHRRReportEntry](#)

[SSoH](#)

[SSoHAttribute](#)

[SSoHR](#)

[SSoHRAttribute](#)

[Standards assignments](#)

[Subheader - SoH mode](#)

[Syntax - message](#)

[System_Health_ID packet](#)

T

[Time_of_Last_Update packet](#)

Timer events

client ([section 3.1.6](#), [section 3.2.6](#))

server ([section 3.1.6](#), [section 3.3.6](#))

Timers

client ([section 3.1.2](#), [section 3.2.2](#))

server ([section 3.1.2](#), [section 3.3.2](#))

[TLV packet](#)

[TLV0_Reserved packet](#)

[TLV1_Reserved packet](#)

[TLVs](#)

[Tracking changes](#)

[Transport - message](#)

Triggered events - higher layer

client ([section 3.1.4](#), [section 3.2.4](#))

server ([section 3.1.4](#), [section 3.3.4](#))

[TV packet](#)

V

[Vendor_Specific packet](#)

[Vendor-extensible fields](#)

[Versioning](#)

W

[Windows behavior](#)