[MS-HCEP]:
Health Certificate Enrollment Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This 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 technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, 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 Open Specifications.

- **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 technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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. The Open Specifications do 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. Certain Open Specifications are 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.

Revision Summary

Date	Revision History	Revision Class	Comments
10/22/2006	0.01		MCPP Milestone 1 Initial Availability
01/19/2007	1.0		MCPP Milestone 1
03/02/2007	1.1		Monthly release
04/03/2007	1.2		Monthly release
05/11/2007	1.3		Monthly release
06/01/2007	1.3.1	Editorial	Revised and edited the technical content.
07/03/2007	1.3.2	Editorial	Revised and edited the technical content.
07/20/2007	2.0	Major	Updated and revised the technical content.
08/10/2007	3.0	Major	Updated and revised the technical content.
09/28/2007	4.0	Major	Updated and revised the technical content.
10/23/2007	5.0	Major	Updated and revised the technical content.
11/30/2007	5.0.1	Editorial	Revised and edited the technical content.
01/25/2008	5.0.2	Editorial	Revised and edited the technical content.
03/14/2008	5.0.3	Editorial	Revised and edited the technical content.
05/16/2008	6.0	Major	Updated and revised the technical content.
06/20/2008	6.1	Minor	Updated the technical content.
07/25/2008	6.1.1	Editorial	Revised and edited the technical content.
08/29/2008	6.1.2	Editorial	Revised and edited the technical content.
10/24/2008	6.1.3	Editorial	Revised and edited the technical content.
12/05/2008	6.1.4	Editorial	Revised and edited the technical content.
01/16/2009	6.1.5	Editorial	Revised and edited the technical content.
02/27/2009	6.1.6	Editorial	Revised and edited the technical content.
04/10/2009	6.1.7	Editorial	Revised and edited the technical content.
05/22/2009	6.2	Minor	Updated the technical content.
07/02/2009	6.2.1	Editorial	Revised and edited the technical content.
08/14/2009	6.3	Minor	Updated the technical content.

Date	Revision History	Revision Class	Comments
09/25/2009	6.4	Minor	Updated the technical content.
11/06/2009	6.4.1	Editorial	Revised and edited the technical content.
12/18/2009	6.5	Minor	Updated the technical content.
01/29/2010	6.6	Minor	Updated the technical content.
03/12/2010	7.0	Major	Updated and revised the technical content.
04/23/2010	7.1	Minor	Updated the technical content.
06/04/2010	7.2	Minor	Updated the technical content.
07/16/2010	8.0	Major	Significantly changed the technical content.
08/27/2010	9.0	Major	Significantly changed the technical content.
10/08/2010	10.0	Major	Significantly changed the technical content.
11/19/2010	11.0	Major	Significantly changed the technical content.
01/07/2011	12.0	Major	Significantly changed the technical content.
02/11/2011	13.0	Major	Significantly changed the technical content.
03/25/2011	14.0	Major	Significantly changed the technical content.
05/06/2011	14.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	15.0	Major	Significantly changed the technical content.
09/23/2011	15.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	16.0	Major	Significantly changed the technical content.
03/30/2012	16.1	Minor	Clarified the meaning of the technical content.
07/12/2012	16.2	Minor	Clarified the meaning of the technical content.
10/25/2012	17.0	Major	Significantly changed the technical content.
01/31/2013	17.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	18.0	Major	Significantly changed the technical content.
11/14/2013	18.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/13/2014	18.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1   Introduction

This document specifies the Health Certificate Enrollment Protocol. The Health Certificate Enrollment Protocol is a remote procedure call (RPC) interface that allows a network endpoint to obtain digital certificates. These certificates are conditionally issued based on the compliance of that endpoint with security policy defined for the network.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1   Glossary

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

Abstract Syntax Notation One (ASN.1)
Active Directory
Active Directory domain
base64
Basic Encoding Rules (BER)
certificate authority (CA)
certificate chain
certification
cryptographic service provider (CSP)
directory
Distinguished Encoding Rules (DER)
domain
enforcement client
enroll/enrollment
extended key usage (EKU)
fully qualified domain name (FQDN)
health certificate
health certificate enrollment agent (HCEA)
health policy server
health registration authority (HRA)
health state
HTTP Internal Server Error
HTTP OK
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Internet Protocol security (IPsec)
object identifier (OID)
Public Key Cryptography Standards (PKCS)
registration authority (RA)
Remote Authentication Dial-In User Service (RADIUS)
Rivest-Shamir-Adleman (RSA)
self-signed certificate
statement of health (SoH)
statement of health response (SoHR)
system health entity
trusted platform module (TPM)

Uniform Resource Locator (URL)
user agent

The following terms are specific to this document:

Cryptographic Application Programming Interface (CAPI): Also known as Windows **Cryptographic Application Programming Interface**, CryptoAPI, and Microsoft Cryptography API. An application programming interface (API) that allows developers who use the Windows operating system to secure Windows-based applications.

PEP channel: An abstract interface that is used by the NAP client to transport **SoH** messages to and from the PEP. Examples of PEP channels are DHCP and HTTP/S used to transport **SoH** messages.

NAP EC API: Provides a set of function calls that allow NAP **enforcement clients** to register with the NAP agent, to request system health status, and to pass system health remediation information to the NAP agent. The **NAP EC API** allows vendors to create and install additional NAP ECs. For more information about this API, see [MS-NAPSO] section 3.3.1 and [MSDN-NAPAPI].

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

1.2   References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [Windows Protocol].

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.

[ITUX680] ITU-T, "Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", Recommendation X.680, July 2002, http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-0207.pdf

[MS-GPNAP] Microsoft Corporation, "Group Policy: Network Access Protection (NAP) Extension".

[MS-NAPSO] Microsoft Corporation, "Network Policy and Access Services System Overview".

[MS-WCCE] Microsoft Corporation, "Windows Client Certificate Enrollment Protocol".

[RFC20] Cerf, V., "ASCII Format for Network Interchange", RFC 20, October 1969, http://www.ietf.org/rfc/rfc20.txt

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

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, March 1998, http://www.ietf.org/rfc/rfc2315.txt

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt

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

[RFC2446] Silverberg, S., Mansour, S., Dawson, F., and Hopson, R., "iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-Dos, and Journal Entries", RFC 2446, November 1998, http://www.ietf.org/rfc/rfc2446.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC2797] Myers, M., Liu, X., Schaad, J., and Weinstein, J., "Certificate Management Messages Over CMS", RFC 2797, April 2000, http://www.ietf.org/rfc/rfc2797.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

[RFC2986] Nystrom, M., and Kaliski, B., "PKCS#10: Certificate Request Syntax Specification", RFC 2986, November 2000, http://www.ietf.org/rfc/rfc2986.txt

[RFC3174] Eastlake III, D., and Jones, P., "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001, http://www.ietf.org/rfc/rfc3174.txt

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt

[RFC3548] Josefsson, S., Ed., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003, http://www.ietf.org/rfc/rfc3548.txt

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004, http://www.ietf.org/rfc/rfc3852.txt

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, http://www.ietf.org/rfc/rfc4559.txt

[TNC-IF-TNCCSPBSoH] TCG, "TNC IF-TNCCS: Protocol Bindings for SoH", version 1.0, May 2007, http://www.trustedcomputinggroup.org/resources/tnc_iftnccs_protocol_bindings_for_soh_version_1_0/

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

1.2.2   Informative References

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

[MS-CERSOD] Microsoft Corporation, "Certificate Services Protocols Overview".

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

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

[MSDN-CAPI] Microsoft Corporation, "Cryptography", http://msdn.microsoft.com/en-us/library/aa380255.aspx

[MSDN-CERTCERTSTR] Microsoft Corporation, "Certificate and Certificate Store Functions", http://msdn.microsoft.com/en-us/library/aa380252(VS.85).aspx#certificate_and_certificate_store_functions

[MSDN-CSP] Microsoft Corporation, "Cryptographic Provider Names", http://msdn.microsoft.com/en-us/library/aa380243.aspx

[MSDN-NAPAPI] Microsoft Corporation, "NAP Interfaces", http://msdn.microsoft.com/en-us/library/aa369705(v=VS.85).aspx

[MSFT-HRA] Microsoft Corporation, "Configure HRA Automatic Discovery", http://technet.microsoft.com/en-us/library/dd296901(WS.10).aspx

[MSFT-IPSEC] Microsoft Corporation, "IPsec", http://technet.microsoft.com/en-us/network/bb531150.aspx

[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

[RFC3447] Jonsson, J., and Kaliski, B., "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003, http://www.ietf.org/rfc/rfc3447.txt

[TPM] Trusted Computing Group, "TPM Work Group", https://www.trustedcomputinggroup.org/groups/tpm/

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

1.3 Overview

Many network administrators maintaining a secure network require that clients accessing their networks comply with policies established for the network. For example, an administrator might require that every client accessing the network have an active firewall. One of the ways that administrators can ensure compliance of network endpoints is to require that clients **enroll** for a **health certificate**. Health certificates encapsulate client compliance with policy in a way that can be presented to interested parties without requiring those parties to perform the validation themselves.

The Health Certificate Enrollment Protocol is designed to accomplish health certificate enrollment. The client sends a Health Certificate Enrollment Protocol request to a **health registration authority (HRA)**. The Health Certificate Enrollment Protocol request includes a certificate request and a report of the client's current **health state**. The HRA communicates with **health policy servers** and **certification authorities** to form and send a Health Certificate Enrollment Protocol response to the client. If the client is compliant, the response contains an issued certificate and the results of the validation of the client's health state against policy.

The Health Certificate Enrollment Protocol has authenticated and unauthenticated modes. In the authenticated mode, the Health Certificate Enrollment Protocol supports authentication of the server, client, or both. Authentication of the server in the Health Certificate Enrollment Protocol is achieved by using the **Hypertext Transfer Protocol (HTTP)** over Transport Layer Security (TLS), as specified in [RFC2818]. During the establishment of the TLS channel, as specified in [RFC2446], the server is authenticated by the client. Authentication of the client in the Health Certificate Enrollment Protocol is achieved by using Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)–based Kerberos and NTLM HTTP authentication, as specified in [RFC4559].

The Health Certificate Enrollment Protocol is typically deployed in an environment such as the one in the following figure. The use of a policy engine as a **Remote Authentication Dial-In User Service (RADIUS)** proxy is an optional configuration in enterprise environments.
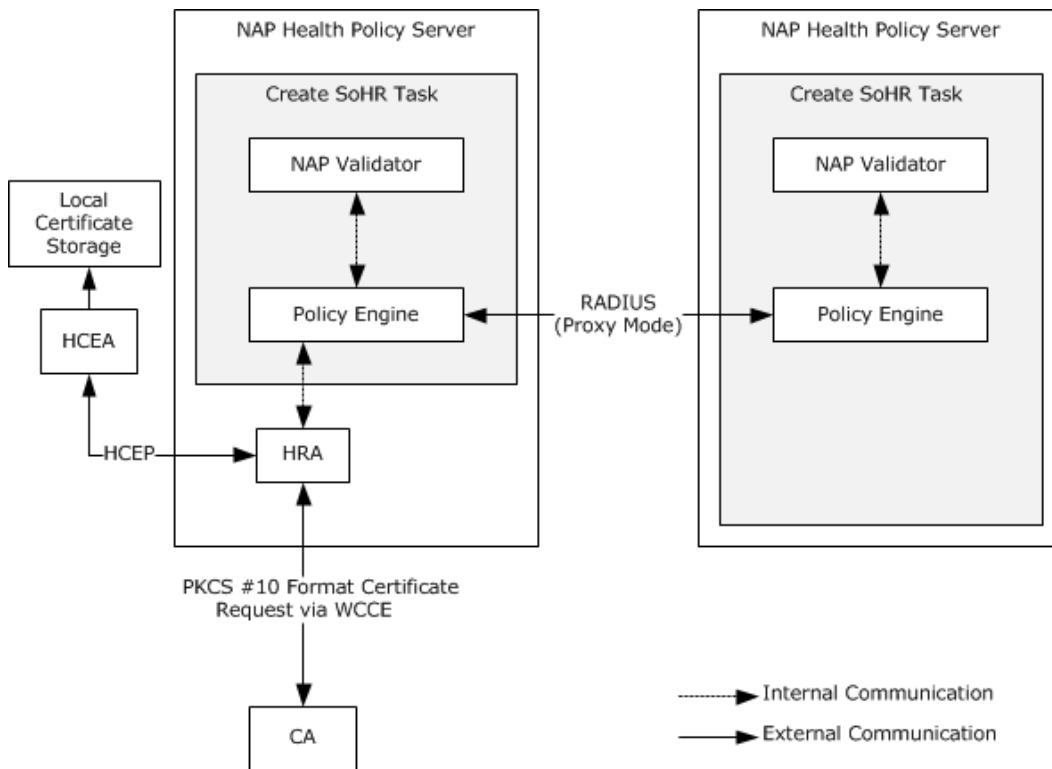

Figure 1: Deployment environment for the Health Certificate Enrollment Protocol

In this example, the flow is as follows:

1. The **health certificate enrollment agent (HCEA)** sends a Health Certificate Enrollment Protocol request to the HRA. The HRA is identified by the **Uniform Resource Locator (URL)** with which the HCEA is provisioned. The protocol that is specified by the provisioned URL determines whether the HRA is authenticated by using TLS. (HTTP over TLS is equivalent to **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS).**)

   The payload of a Health Certificate Enrollment Protocol request message, sent by the HCEA, contains a **Public Key Cryptography Standards (PKCS)** #10 certificate request (as specified in [RFC2986]). The PKCS #10 request contains a **Statement of Health (SoH)** message, as specified in [TNC-IF-TNCCSPBSoH] section 3.5.

2. The HRA sends the SoH to a health policy server for evaluation as per network policies either on an internal interface or over any standard or implementation specific protocol. In the example depicted in the preceding figure showing the deployment environment for the Health Certificate Enrollment Protocol, Remote Authentication Dial-In User Service (RADIUS) is used (for more information, see [RFC2865]) using Microsoft RADIUS Attributes for Network Access Protection (for more information, see [MS-RNAP]). The health policy server determines the health state of the client and informs the HRA of this.

3. If the health state of the client is compliant, the HRA requests a certificate authority (CA) to issue a certificate. The Microsoft implementation of the HRA uses the Windows Client Certificate Enrollment Protocol, as specified in [MS-WCCE], to request and receive the certificate. If the client's health state is not compliant, the HRA MAY still request a certificate from the certificate authority with the certificate containing an indication that the client is unhealthy.

4. The HRA sends a Health Certificate Enrollment Protocol response.

   If the health state of the client is compliant, the payload of the response contains a **statement of health response (SoHR)** (as specified in [TNC-IF-TNCCSPBSoH]) and a PKCS #7 message (as specified in [RFC2315]) with an X.509 certificate (for more information, see [X509]), as specified in [RFC3280].

   If the health state of the client is not compliant, the payload of the response contains an SoHR and MAY contain a PKCS #7 message containing the certificate.

5. The HCEA receives the HCEP response and, if the certificate was included, deposits the certificate in the **PersistedComputerCertificates** ADM element specified in section 3.1.1.

1.4   Relationship to Other Protocols

The Health Certificate Enrollment Protocol uses HTTP (as specified in [RFC2616]) or HTTP over TLS (as specified in [RFC2818]) as the transport for its messages. The payload of an HCEP request message that is sent by the HCEA, contains a PKCS #10 certificate request (as specified in [RFC2986]), which contains an SoH message (as specified in [TNC-IF-TNCCSPBSoH] section 3.5).

If the client's health state is compliant, the health registration authority (HRA) requests a certificate authority (CA) to issue a certificate. The Microsoft implementation of the HRA uses the Windows Client Certificate Enrollment Protocol [MS-WCCE] to request and receive the certificate; other implementations can choose different means of communication between HRA and CA. If the client's health state is not compliant, the HRA MAY still request a certificate from the certificate authority, with the certificate containing an indication that the client is unhealthy.

The HRA sends an HCEP response, the payload of which contains an SoHR, as specified in [TNC-IF-TNCCSPBSoH] section 3.6, and if the client is compliant with health policies, it also includes a PKCS #7 message (as specified in [RFC2315]) with possibly an X.509 certificate, as specified in [RFC3280].
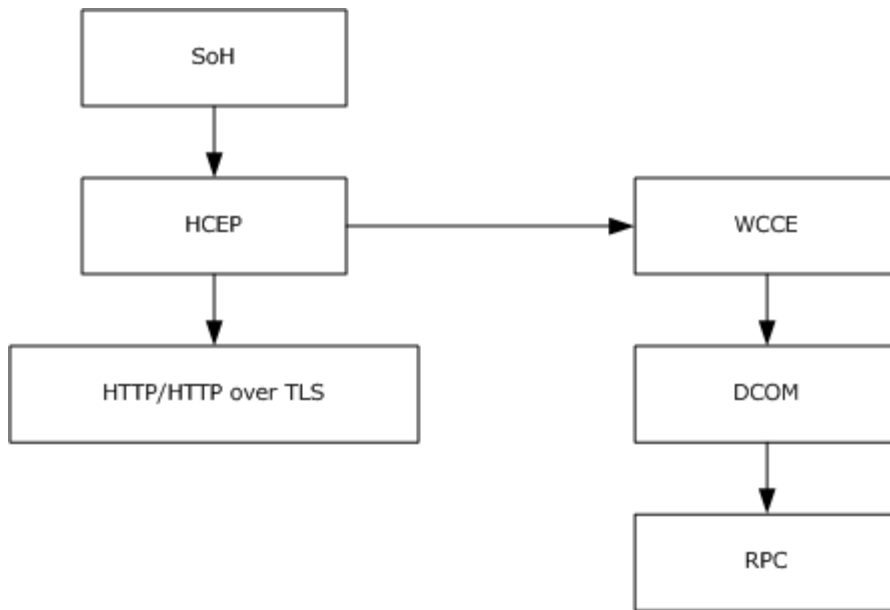
Figure 2: Diagram illustrating the relationship between HCEP and other protocols

1.5 Prerequisites/Preconditions

For Health Certificate Enrollment Protocol communication to begin, the prerequisite configuration for HCEA is as follows:

1. The HCEA has to be configured with the list of the URLs of the HRAs via an implementation-dependent method.<1>The protocol that is specified by the specific provisioned URL determines if the respective HRA is authenticated by using TLS. (HTTP over TLS is equivalent to HTTPS.) For more information, see section 1.3.

2. The HCEA has to be configured with the required security parameters to construct a certificate request that is sent in the Health Certificate Enrollment Protocol request. These include, but are not limited to:

   ▪ The algorithm and key length of the public-private key pair associated with the certificate. The HCEA has to support the **Rivest-Shamir-Adleman (RSA)** algorithm with a key length of 2,048 bits. Other key lengths can<2> be supported. For more information, see [RFC3447].

   ▪ The signature algorithm used to sign the certificate request. The HCEA has to support the Secure Hash Algorithm 1 (SHA1), as specified in [RFC3174].

   It is preferable for the HCEA and HRA to agree on the algorithm that the Health Certificate Enrollment Protocol uses.<3>

3. The HCEA and HRA implementations are required to agree on the same **object identifier (OID)** values for the fields in section 2.2.1.4.<4>

The prerequisite configuration for the HRA is the following:

▪ If the HRA is configured to authenticate the client, the settings required, as specified in [RFC4559] section 4.1, have to be configured on the HRA.

If the HRA is configured to authenticate the client, the handling of the authentication by the HTTP layer as specified in [RFC4559] is a precondition for processing the HCEP request. The failure of the authentication has to be processed by HTTP in accordance with [RFC4559].

1.6 Applicability Statement

The Health Certificate Enrollment Protocol allows a client machine to obtain an X.509 certificate, as specified in [RFC3280], that represents its compliance to policy. Because the Health Certificate Enrollment Protocol relies on the client to make accurate reports of its current state, the protocol is not applicable by itself in environments where compliance of the client must be absolutely guaranteed. However, the Health Certificate Enrollment Protocol MAY<5> be used in such environments if supplemented by the use of hardware credentials or other suitable security mechanisms (for more information, see [TPM]) that can improve the reliability of the client reports.

Applicable uses of such an X.509 certificate include, but are not limited to, certificate-based **Internet Protocol security (IPsec)**, as specified in [RFC2409]. In an IPsec scenario, network administrators can require clients to comply with the network security policies before accessing resources on the network. For example, administrators can configure IPsec policies to require a client to present a health certificate to the resource (as an indication of the client compliance with network security policies) before the client can have access to the resource.

1.7 Versioning and Capability Negotiation

The Health Certificate Enrollment Protocol does not perform any version detection or capability negotiation by itself since this is the first version.

Although this version of the Health Certificate Enrollment Protocol does not support the concept of versioning, there is a version number field in Health Certificate Enrollment Protocol messages. This field is intended for future use. In the current version of the Health Certificate Enrollment Protocol, implementations are required to set the version field value to "1.0". The fixed value version field (see sections 2.2.1.2 and 2.2.2.2) is intended to enable future versions of the protocol to negotiate a version that is commonly supported by both HCEA and HRA.

1.8 Vendor-Extensible Fields

There are no vendor-extensible fields for this protocol.

1.9 Standards Assignments

Microsoft has been assigned the object identifier (OID) 1.3.6.1.4.1.311 by the Internet Assigned Numbers Authority (IANA), as the Microsoft private enterprise IANA code. (For more information, see [IANA-ENT].)

The Windows implementation of the Health Certificate Enrollment Protocol uses the extension 1.3.6.1.4.1.311.47.1.1 under this OID for identifying the health extension for certificates and certificate requests, as specified in section 2.2.1.4.

2   Messages

2.1   Transport

Health Certificate Enrollment Protocol messages MUST be transmitted over HTTP, as specified in [RFC2616], or HTTP over TLS, as specified in [RFC2818]. HCEP MUST be encapsulated within these protocols as a POST method (as specified in [RFC2616] section 9.5) to a specific URL. (The existence of the URL is a precondition of the protocol, as specified in section 1.5). The POST MUST have specific HTTP headers and an entity body (as specified in [RFC2616]) formatted as specified later in section 2.2.1.

The Health Certificate Enrollment Protocol is encapsulated within and depends on HTTP or HTTP over TLS for delivery of messages. The Health Certificate Enrollment Protocol does not impose any message retransmissions or other requirements on those transports. The choice of the transport is based entirely on the URL present at the HCEA. For more information about the configuration of the HCEA, see section 1.5.

2.2   Message Syntax

All strings that follow are ASCII strings, as specified in [RFC20], and MUST conform to the general HTTP rules on string values in headers, as specified in [RFC2616] section 4.2.

2.2.1   HCEP Request

The HCEP request is an HTTP POST request that is made to a provisioned URL.

The HCEP request that is encapsulated in HTTP includes a number of fields in the HTTP message header. Some of them are standard fields (as specified in [RFC2616] sections 4.5, 5.3, and 7.1) that must take on specific values; however, others are new fields that are defined by the Health Certificate Enrollment Protocol. These fields MUST follow the rules as specified in [RFC2616] section 4.2.

2.2.1.1   Standard HTTP Message Header Fields

The HTTP message headers in the HCEP request MUST include the following fields with these specified values.

Tokens

Pragma: MUST be "no-cache". This is an HTTP header, as specified in [RFC2616] section 4.5.

Content-Type: MUST be "application/healthcertificate-request". This is an HTTP header field, as specified in [RFC2616] section 7.1.

Content-Length: MUST be the size in bytes of HTTP message body (as specified in section 2.2.1.3). This is an HTTP header field, as specified in [RFC2616] section 7.1.

The HTTP message header MAY<6> contain other fields, as specified in [RFC2616]. These fields MAY<7> be ignored by the HRA during the processing of the HTTP header. An example of such a field is the **User-Agent** field, as specified in [RFC2616] section 14.43:

▪ User Agent: "NAP IPSec Enforcement v1.0".

2.2.1.2 HTTP Message Header Fields Introduced by HCEP

The HTTP message headers in the HCEP request MUST include the following fields.

Tokens

HCEP-Version: MUST be a string with value "1.0".

HCEP-Correlation-Id: This MUST be a Correlation Id that is **base64**-encoded (as specified in [RFC3548] section 3). A Correlation Id (as specified in [TNC-IF-TNCCSPBSoH] section 3.8.6) is a 24-byte value that uniquely identifies a Health Certificate Enrollment Protocol transaction originating from the HCEA. It SHOULD<8> be the same as the Correlation Id that is present in SoH, as specified in [TNC-IF-TNCCSPBSoH] section 3.8.

2.2.1.3 HTTP Message Body Used in an HCEP Request

The HCEP request MUST contain an HTTP message body, which follows the rules specified in [RFC2616] section 4.3. The value of this message body MUST be an **ASN.1 DER**–encoded health certificate request. The format of the health certificate request MUST be as specified in section 2.2.1.4.

2.2.1.4 Health Certificate Request

The health certificate request MUST be in PKCS #10 format (as specified in [RFC2986]) and be encoded in ASN.1 Distinguished Encoding Rules (DER), as specified in [ITUX680]. This MUST be present in an HCEP request. The HRA processing of the health certificate request is specified in section 3.2.5. It MUST contain the following parts, as specified in [RFC2986] section 4.

Tokens

subject: SHOULD be a zero-length string if the **ClientAuthenticationFlag** ADM element specified in section 3.1.1 is TRUE; otherwise, it SHOULD be the predefined string "Anonymous System Health Authentication", as specified in [RFC3280] section 4.1.2.6.<9>

subjectPublicKeyInfo: MUST be a public key for the X.509 certificate, as specified in [RFC3280] section 4.1.

PKCS #10 Attributes: The following X.509 certificate extensions are encapsulated as PKCS #10 attributes in the health certificate request. PKCS #10 attributes are as specified in [RFC2986] section 4.1.

Extended Key Usage: A health certificate request MUST contain an **Extended Key Usage (EKU)** extension (as specified in [RFC3280] section 4.2.1.1) with the OID values specified as follows:

- MUST have an OID to indicate that the certificate request is a health certificate request.<10>

- SHOULD have the OID value that indicates id-kp-clientAuth, as specified in [RFC3280] section 4.2.1.13, if the **ClientAuthenticationFlag** ADM element specified in section 3.1.1 is TRUE.<11>

Subject Alternative Name: MUST be the **fully qualified domain name (FQDN)** of the client if the **ClientAuthenticationFlag** ADM element specified in section 3.1.1 is TRUE. This extension MUST be as specified in [RFC3280] section 4.2.1.7.

Statement of Health Certificate Extension: The certificate request MUST have a certificate extension, as specified in [RFC3280]. This certificate extension MUST contain the ASN.1 DER–encoded statement of health data, as specified in section 2.2.3.7.

Cryptographic Service Provider Certificate Extension: The certificate request MUST have a certificate extension to specify the name of the **cryptographic service provider (CSP)** used to generate the key pair on the HCEA. This attribute contains the name of the CSP that is used to generate the key pair on the HCEA. The extension and the OID value MUST be as specified in [MS-WCCE] section 2.2.2.7 that defines the certificate request attributes.<12>

The certificate request MUST be signed to prevent tampering by using one of the preconfigured signature algorithms specified in section 1.5. The signature over the certificate request MUST be included in the PKCS #10 message, as specified in [RFC2986].

2.2.2 HCEP Response

The HCEP response MUST be an **HTTP OK** response (status-code 200), indicating that there are no errors. Other HTTP response status codes, as specified in [RFC2616] section 6.1.1, MUST be returned by the server in case of error events, as specified in section 3.1.8. The client-side handling of errors is specified in section 3.1.8.

The HCEP response that is encapsulated in HTTP includes a number of fields in the HTTP message header. Some are standard fields (as specified in [RFC2616] sections 4.5, 6.2, and 7.1) that must take on specific values; however, others are new fields that are defined by the Health Certificate Enrollment Protocol. These fields MUST follow the rules as specified in [RFC2616] section 4.2.

2.2.2.1 Standard HTTP Message Header Fields

The HTTP message headers in the HCEP response MUST include the following fields.

Tokens

Cache-Control: MUST be "no-cache, must-revalidate" or "must-revalidate, no-cache". This is an HTTP header, as specified in [RFC2616] section 4.5.

Content-Type: MUST be "application/healthcertificate-response". This is an HTTP header field, as specified in [RFC2616] section 7.1.

Content-Length: MUST be the size of ASN.1 DER–encoded health certificate response.

2.2.2.2 HTTP Message Header Fields Introduced by HCEP

The HTTP message headers in the HCEP response MUST include the following Tokens.

Tokens

HCEP-Version: MUST be a string with value "1.0".

HCEP-Correlation-Id: This MUST be a Correlation Id that is base64-encoded (as specified in [RFC3548] section 3). A Correlation Id (as specified in [TNC-IF-TNCCSPBSoH] section 3.8.6) is a 24-byte value that uniquely identifies a Health Certificate Enrollment Protocol transaction originating from the HCEA. It SHOULD<13> be the same as the Correlation Id that is present in SoH, as specified in [TNC-IF-TNCCSPBSoH] section 3.8.

HCEP-SoHR: MUST be the base64-encoded statement of health response (SoHR), as specified in [TNC-IF-TNCCSPBSoH] section 3.6.

The SoHR is the result of the validation of the SoH, as specified in [TNC-IF-TNCCSPBSoH] section 3.5, received by the HRA from the HCEP request. It is an opaque sequence of bytes that is not interpreted or used directly by the Health Certificate Enrollment Protocol.

HCEP-AFW-Protection-Level: MUST be a string that is specified as follows:

This value MUST be sent back from the HRA in an HCEP response. Valid values for this field MUST be:

- "1" to indicate that the certificate payload in the HCEP response can be used for signing data.

- "2" to indicate that the certificate payload in the HCEP response can be used for signing and encrypting data.

HCEP-AFW-Zone: MUST be a string specified as follows:

This field MUST be sent back from the HRA in an HCEP response. Valid values for this field MUST be the ASCII (as specified in [RFC20]) representation of the decimal form of integers from zero to 2^32 -1 inclusive. This is used as a hint for dynamic selection of a preconfigured policy by the consumer of the health certificate on the client.<14>

The header MAY<15> contain other fields besides those listed here. All other fields SHOULD be ignored by the Health Certificate Enrollment Protocol client.

2.2.2.3   HTTP Message Body Used in an HCEP Response (HTTP OK Response)

The health certificate response MUST be present if the client is deemed to be compliant with policy as the HTTP message body, which follows the rules as specified in [RFC2616] section 4.3.<16> If the message body is present, the value of this message body MUST be an ASN.1 DER–encoded health certificate response. The format of the health certificate response MUST be as specified in section 2.2.2.4.

2.2.2.4   Health Certificate Response

The health certificate response MUST be in PKCS #7 format (as specified in [RFC2315]) and be encoded in ASN.1 DER, and contain the issued X.509 certificate (as specified in [RFC3280]) in a **certificate chain**. The format of this response MUST be as specified in [MS-WCCE] section 2.2.2.8.

The health certificate response MUST be present if the client is deemed to be compliant with policy.

The health certificate response MAY be present if the client is deemed to be noncompliant with policy.<17>

2.2.3   Certificate Request OIDs

This section defines the symbolic names of OIDs used in the "Creating Health Certificate Request by HRA" section 3.2.5.4.

2.2.3.1   napPolicyInformationCompliantOid

OID = 1.3.6.1.4.1.311.47.1.10

Description: Used to indicate that the client is compliant with health policy.

Format: Specified in [RFC3280] section 4.2.1.5.

2.2.3.2 napPolicyInformationNotCompliantOid

OID = 1.3.6.1.4.1.311.47.1.11

Description: Used to indicate that the client is not compliant with health policy.

Format: Specified in [RFC3280] section 4.2.1.5.

2.2.3.3 napPolicyInformationIsolationStateOid

OID = 1.3.6.1.4.1.311.47.1.12

Description: when used as an identifier in a policy structure (either policy information or application policy), this OID indicates the structure contains additional information regarding the client's isolation state.

The isolation state is one of the following values:

- Compliant.

- Network connectivity is not being restricted but might be at a later time.

- Noncompliant.

Format: Specified in [RFC3280] section 4.2.1.5.

2.2.3.4 napPolicyInformationExtendedStateOid

OID = 1.3.6.1.4.1.311.47.1.13

Description: When used as identifier in a policy structure (either policy information or application policy), this OID indicates the structure contains additional information on the client state.

The extended state is one of the following values:

- No additional data.

- Transition data.

- Infected data.

- Unknown data.

Format: Specified in [RFC3280] section 4.2.1.5.

2.2.3.5 napHealthyOid

OID = 1.3.6.1.4.1.311.47.1.1

Description: When used as an extended key usage ([RFC3280] section 4.2.1.13) or Application Policy ([MS-WCCE] section 2.2.2.7.7.3), this OID indicates that the client is healthy.

2.2.3.6 napUnhealthyOid

OID = 1.3.6.1.4.1.311.47.1.3

Description: When used as an extended key usage ([RFC3280] section 4.2.1.13) or Application Policy ([MS-WCCE] section 2.2.2.7.7.3), this OID indicates that the client is unhealthy.

2.2.3.7   napSoHOid

OID = 1.3.6.1.4.1.311.47.1.1

Description: SoH message (specified in [TNC-IF-TNCCSPBSoH]) sent from the NAP client in the "Statement of Health Certificate Extension" specified in section 2.2.1.4.

Format: The following is the ASN.1 format for this attribute:

```
SoHExtension ::= OCTET STRING
```

3   Protocol Details

The Health Certificate Enrollment Protocol is a simple request-response protocol. The Health Certificate Enrollment Protocol allows a network endpoint to obtain digital certificates. These certificates are conditionally issued based on the compliance of that endpoint with security policy defined for the network.<18>

The health certificate enrollment agent (HCEA) sends a Health Certificate Enrollment Protocol request and the HRA responds with a Health Certificate Enrollment Protocol response. The protocol is used by the HCEA to obtain a health certificate based on its compliance with the security policies defined for the network. The protocol is always a single Health Certificate Enrollment Protocol request followed by a single Health Certificate Enrollment Protocol response, as shown in the following diagram.
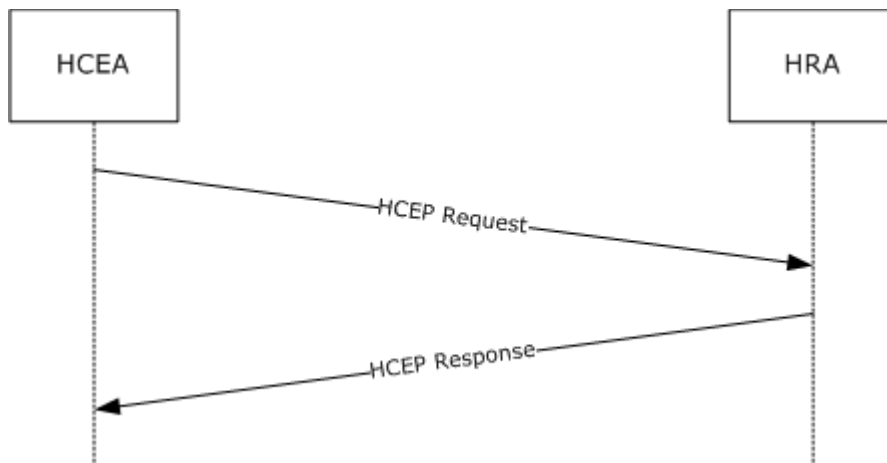

Figure 3: Health Certificate Enrollment Protocol single request and response

3.1   Client Details

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

CorrelationId: An HCEP-Correlation-Id, as specified in section 2.2.1.2. The HCEA SHOULD use this ADM element to store the HCEP-Correlation-Id that it sent in its HCEP request.

ClientAuthenticationFlag: A Boolean flag. The HCEA SHOULD<19> maintain this flag to specify if it expects the HRA to authenticate the client, as specified in [RFC4559]. This flag determines the values of the **subject**, **Subject Alternative Name**, and **Extended Key Usage** fields in the health certificate request, as specified in section 2.2.1.4.

PersistedComputerCertificates: A repository where health certificates reside. The client uses this repository to store health certificates that have been received from the server during an HCEP response (section 2.2.2).<20>

3.1.2   Timers

None.

3.1.3   Initialization

The HCEA constructs and sends an HCEP request, as specified in section 2.2.1.1. If the transport that the provisioned URL specifies is HTTP over TLS (as specified in [RFC2818]), before communication begins, the client MUST be set up to trust the certificate authority that is issuing the server certificate.

3.1.4   Higher-Layer Triggered Events

The following higher-layer events affect a Health Certificate Enrollment Protocol client's operation:

- Any state change that can affect the client compliance to policy SHOULD trigger the HCEA to re-enroll for a health certificate from the configured HRA.<21>

- Any configuration state change that changes the HRA URL configuration SHOULD trigger the HCEA to re-enroll for a health certificate from the configured HRA.<22>

The higher layer SHOULD use the **INapEnforcementClientCallback::NotifySoHChange** call, which is part of the **NAP EC API**, to trigger the HCEP to start a new enrollment.

3.1.5   Message Processing Events and Sequencing Rules

3.1.5.1   Sending an HCEP Request

HCEA MUST create an HCEP request as follows:

1. Create a health certificate request:

    1. MUST set the subject, subjectPublicKeyInfo, EKU, and Subject Alternative Name in the PKCS #10 (as specified in [RFC2986]) part of the request, as specified in section 2.2.1.4.

    2. MUST obtain an SoH, as specified in [TNC-IF-TNCCSPBSoH], from the **system health entity**, and set the SoH certificate extension as specified in section 2.2.1.4. Obtaining the SoH SHOULD be done by calling **INapEnforcementClientBinding::GetSoHRequest**, which is part of the NAP EC API, and then calling the **INapEnforcementClientConnection::GetSoHRequest**, which is part of the NAP EC API, to retrieve the SoH as an opaque buffer.

2. MUST set the HTTP message body to an ASN.1 DER (as specified in [ITUX680]) health certificate request as specified in section 2.2.1.

3. Set the HTTP message headers as follows:

    1. MUST set the **Pragma**, **Content-Type**, **HCEP-Version** and the **Content-Length** header fields, as specified in section 2.2.1.

    2. MUST set the HCEP-Correlation-Id, as specified in section 2.2.1.2. It MUST be base64-encoded, as specified in [RFC3548]. The value SHOULD be stored in the **CorrelationId** ADM element as specified in section 3.1.1.

4. The HCEA MUST then select the provisioned URL from the list of HRA URLs specified by section 1.5 and send the request over the transport designated by the scheme (see [RFC2396] for the

definition of the scheme part and [RFC2616] and [RFC2818] for designation of HTTP and HTTP over TLS schemes, respectively) to the HRA. If the HRA is configured to authenticate clients as specified in section 1.5, the authentication will be handled by the transport as specified by [RFC4559].

5. If the transport returns an error in either transmitting or receiving the response, HCEA MUST select the next URL from the provisioned list and repeat step 4.

6. If the HRA returned the response, HCEA MUST proceed to "Processing an HCEP Response" as specified in section 3.1.5.2

7. If attempts to work with all provisioned URLs in steps 4 and 5 returned an error, HCEA MUST handle the condition as specified in section 3.1.8.

3.1.5.2   Processing an HCEP Response

If the HCEA receives an HTTP response other than HTTP OK, it MUST perform the error handling as specified in section 3.1.8. On receiving an HTTP OK response, HCEA MUST perform the validation as follows:

1. HCEA MUST validate that all the HCEP response header fields are present in the response, as specified in section 2.2.1.2, and MUST discard the response if the validation fails.

2. HCEA MAY<23> validate the values of all the HCEP response header fields, as specified in sections 2.2.2.1 and 2.2.2.2 and MUST discard the response if the validation fails.

3. HCEA SHOULD <24>compare the value of the HCEP-correlation-Id header to the value stored in the **CorrelationId** ADM element as specified in section 3.1.1; if the values do not match, HCEA MAY <25> discard the HCEP response.

4. HCEA MUST validate the format of the health certificate response that is present in the HCEP response, as specified in section 2.2.2.4. If the health certificate response is invalid, the HCEA MUST discard the HCEP response.

5. When the validation checks are complete, the HCEA processes the HCEP response and MUST extract the base64-encoded SoHR from the HCEP-SoHR HTTP header (which is described in section 2.2.2.2) and decode it. The decoded SoHR MUST be passed to the system health entity, which SHOULD be done by calling **INapEnforcementClientConnection::SetSoHResponse**, which is part of the NAP EC API, and passing the SoHR as an opaque buffer; and then calling the **INapEnforcementClientBinding::ProcessSoHResponse**, which is part of the NAP EC API.

6. HCEA MUST extract the certificate and place it in the **PersistedComputerCertificates** ADM element specified in section 3.1.1. If the specific implementation of this ADM element supports associating metadata with the certificate, the HCEA SHOULD store values of **HCEP-AFW-Zone** and **HCEP-AFW-Protection-Level** headers of the HCEP Response in the metadata associated with the certificate. The certificate is shared with the intended consumers (see section 1.6) by means of the **PersistedComputerCertificates** ADM element. The intended consumers SHOULD use the specific implementation of ADM in order to retrieve the certificate. <26>

3.1.6   Timer Events

None.

3.1.7 Other Local Events

If the IP address of the client machine changes, a new network might be reachable. The client MAY<27> choose to send an HCEP request to a server with which a previous request was unsuccessful.

3.1.8 Client-Side Error Handling

After creating an HCEP request, if the HCEA encounters any errors (including HTTP errors as specified in [RFC2616] section 10), either during transmission of the HCEP request or while waiting for an HCEP response, it MUST discard the HCEP request. The only exception is when an HTTP error is encountered during transmission resulting from a redirect or authentication challenge, thus allowing the retransmission of the original HCEP-Correlation-Id value.

Any subsequent requests by the client MUST have a new HCEP-Correlation-Id value.

3.2 Server Details

3.2.1 Abstract Data Model

HRA MUST store the HCEP-Correlation-Id that it receives in an HCEP request. It MUST use the same HCEP-Correlation-Id in the HCEP response that it generates.

The following is a list of predefined settings that SHOULD<28> be present on the server and SHOULD be initialized by the server administrator:

- A list of strings that are used to restrict the requests based on the user-agent strings, as specified in section 2.2.2.1, that are present in an HCEP request. If no string in the list is a substring of the user-agent string, the request MUST be discarded. If this list is empty, all **user agents** SHOULD<29> be allowed.

- A list of algorithms that are used to restrict the allowed public keys in the certificate request. If the list is empty, all algorithms SHOULD<30> be allowed.

- A list of algorithms that are used to restrict the allowed signatures in the certificate request. If the list is empty, all algorithms SHOULD<31> be allowed.

- A list of cryptographic service providers (CSPs) that are used to restrict the CSPs that created the certificate requests. If the list is empty, certificate requests from all CSPs SHOULD<32> be allowed.

- A setting that specifies the maximum size, in kilobytes, of the HCEP request, including the HTTP headers and the body. If the size of the request is less than the maximum size allowed, the HCEP request SHOULD<33> be allowed.

- **CAToConnect:** The connection information of a CA. The fields of the **CAToConnect** structure are as follows:

  - Name: A null-terminated Unicode string that specifies the Name of the CA.

  - EndPoint: A null-terminated Unicode string that specifies the CA server address. The string can be in the form of an HTTPS URL, a fully qualified domain name (FQDN) of the CA, or an IPv4 address of the CA.

- **remote CA name List:** A list of **CAToConnect** ADM elements (specified in this section 3.2.1) that contains the CAs HRA will attempt to obtain health certificates from. For initialization information about **remote CA name List**, see section 3.2.3.

- **CAResponseTimeOut:** A DWORD containing the timeout interval in seconds for a CA response. Values range between 0x00000000 and 0x0000012C, including 0x00000000 and 0x0000012C.

- **EvaluationTimeOut:** A DWORD containing the time the HRA server will wait for the evaluation result in milliseconds.<34>

3.2.2 Timers

The HRA server uses the **CAConnectionDuration** timer to ensure the response time from the CA does not exceed the duration set in the **CAResponseTimeOut** ADM element defined in section 3.2.1. The **CAConnectionDuration** timer is specified per HCEP request connection and is used in the processing steps for obtaining a health certificate from the CA as specified in section 3.2.5.2 as part of the sequence of steps for creating the HCEP response.

The HRA server uses the **SoHEvaluationDuration** timer to ensure the evaluation time does not exceed the duration set in the **EvaluationTimeOut** ADM element defined in section 3.2.1. The **SoHEvaluationEuration** timer is specified per HCEP request connection and is used in the steps for processing an HCEP request as specified in section 3.2.5.2.

3.2.3 Initialization

The **remote CA name List** ADM is initialized during HRA server startup. The **remote CA name List** contains the CAs HRA will attempt to obtain health certificates from. It is configured by the IT admin and stored in persistent storage. Upon HRA server startup the configuration is read from the file and set in the **remote CA name List** in the order it was saved to the file. For example the last CA saved to the file will be the last in the list.

The **CAResponseTimeOut** ADM is initialized during HRA server startup. It is configured by the IT admin and stored in persistent storage. If the value is not set by the IT admin the **CAResponseTimeOut** ADM will be set to 20 seconds.

3.2.4 Higher-Layer Triggered Events

SetSoHR Abstract Interface: An abstract interface used by the Policy Engine to send the SoHR to the HRA. The interface is implemented by the HRA and is defined as follows:

```
HRESULT SetSoHR([in] SoHR message, [in] BOOL compliant,
[in] DWORD AFW-Zone, [in] DWORD AFW-Protection-Level);
```

The SoHR is defined in [TNC-IF-TNCCSPBSoH] section 3.6.

The received SoHR is used for both the Enforce NAP policy task (specified in [MS-NAPSO] section 11 and detailed in [MS-HCEP] section 3.2.5.4) and for Proxy SoHR task (specified in [MS-NAPSO] section 12 and detailed in [MS-HCEP] section 3.2.5.2).

3.2.5   Message Processing Events and Sequencing Rules

3.2.5.1   Validating an HCEP Request

When an HCEP request arrives from the client on the HRA, the HRA MUST perform correctness checks on the request.

The HRA MUST perform the following checks:

1. All the required fields in an HCEP request MUST be present in the received request, as specified in section 2.2.1, including the health certificate request with statement of health certificate extension as specified in section 2.2.1.4.

2. If the HRA is configured to authenticate the client (as specified in section 1.5), it MUST validate that the Subject Alternative Name (as specified in section 2.2.1.4) is the same as the FQDN of the authenticated client.

3. If the HRA is configured to not authenticate the client (as specified in section 1.5), it MUST validate that the Subject Alternative Name (specified in section 2.2.1.4) is empty.

4. The health certificate request that is present in the HCEP request MUST conform with the specification in section 2.2.1.4 for the rest of the fields not previously mentioned.

If any of these checks fail, the HRA MUST respond with an **HTTP Internal Server Error**, as specified in [RFC2616] section 6.1.1.

The HRA SHOULD<35> perform the following checks:

1. The size of the request is less than the maximum size, as specified in section 3.2.1.

2. The user agent that is present in the HCEP request contains one of the strings from the list of strings specified in section 3.2.1.

3. The algorithm that is used for signing the certificate request is present in the list of algorithms specified in section 3.2.1.

4. The public key in the certificate request is created by using an algorithm that is present in the list of algorithms specified in section 3.2.1.

5. The CSP that is used to create the health certificate request is acceptable. This is accomplished by checking that the CSP certificate extension of the health certificate request (as specified in section 2.2.1.4) specifies a CSP that is present in the list of CSPs specified in section 3.2.1.

6. If the HRA is configured to authenticate the client (as specified in section 1.5), it MAY validate the subject and Extended Key Usage as specified in section 2.2.1.4.

7. If the HRA is configured to not authenticate the client (as specified in section 1.5), it MAY validate the subject as specified in section 2.2.1.4.

If any of the previous checks fail, the HRA MUST respond with an HTTP Internal Server Error, as specified in [RFC2616] section 6.1.1.

3.2.5.2   Processing an HCEP Request

After the validation checks are complete, the HRA MUST process the HCEP request as follows:

1. Extract the HCEP-Correlation-Id from the HCEP request, for use when creating the HCEP response.

2. Extract the health certificate request (as specified in section 2.2.1.1) from the HCEP request.

3. Extract the SoH specified in section 2.2.1.4 from the health certificate request by reading the ASN.1 DER–encoded SoH from the health certificate request and decoding it.

4. Initiate the **SoHEvaluationDuration** timer specified in section 3.2.2 and send the SoH for evaluation to the policy server, passed as a parameter to the Proxy SoH task (as specified in [MS-NAPSO] section 7). Passing the SoH to the Proxy SoH Task triggers both that task and three subsequent tasks; Receive SoH, Process SoH and Create and Send SoHR with HCEP as the PEP channel. Once the Create and Send SoHR is completed, a SoHR message is returned along with the compliance decision to the HRA.

   Note The specific implementation will define the technology and the protocols used to communicate between the HRA and the policy server.

   ▪ If there was a failure in any one of the [MS-NAPSO] tasks mentioned in step 4 or the policy server failed to process the SoH, the HCEP Request processing MUST be aborted and HRA MUST respond with an error as specified in section 3.2.8.

   The policy server returns results by invoking **SetSoHR** abstract interface (See section 3.2.4) the *message* parameter is the **SoHR** and the *compliant* parameter is the **compliance decision**. The *AFW-Zone* and *AFW-Protection-Level* parameters are the **Zone** and **Protection Level** used in the creation of the HCEP Response (see section 3.2.5.3).

5. The HCEP response MUST then be created and sent as specified in section 3.2.5.3.

3.2.5.3 Creating and Sending an HCEP Response

1. The HCEP response MUST be created as follows. (The logical values of the headers are specified. The values MUST be formatted and encoded as specified in sections 2.2.2.1 and 2.2.2.2.)

   ▪ Set the value of the **HCEP-Correlation-Id** header to the value of the **HCEP-Correlation-Id** header in the HCEP request.

   ▪ Set the value of the **HCEP-SoHR** header to the *SoHR* parameter of the **SetSoHR** abstract interface.

   ▪ Set the value of the **HCEP-AFW-Zone** header to the *Zone* parameter of the **SetSoHR** abstract interface.

   ▪ Set the value of the **HCEP-AFW-Protection-Level** header to the *Protection Level* parameter of the **SetSoHR** abstract interface.

   ▪ Set the constant value of the **HCEP-Version** header as specified in section 2.2.2.2.

   ▪ Set the constant values of the **Content-Type** and **Cache-Control** headers as specified in section 2.2.2.1.

   ▪ If the **compliance decision** was positive, the HRA MUST request a health certificate from a CA. The HRA SHOULD communicate with the CA using the [MS-WCCE] protocol.<36> The technology and the protocols used for communication between HRA and the CA are implementation specific.

If the **compliance decision** was negative, the HRA can request a health certificate from the CA.<37>

The processing steps for obtaining a health certificate from the CA are:

1. HRA reads the **remote CA name List** ADM element specified in section 3.2.1 from the beginning, and gets the first **CAToConnect** ADM element specified in section 3.2.1.

2. HRA constructs the request to the CA (as specified in section 3.2.5.4).<38>

3. HRA initiates the **CAConnectionDuration** timer specified in section 3.2.2.

4. HRA submits a certificate request by invoking the local event, "submitting certificate request", described in [MS-WCCE] section 3.1.1.6.2, with the following parameters:

   - *CAName*: The value of the CAToConnect.Name.

   - *ServerName*: The value of the CAToConnect.EndPoint.

   - *Flags*: The flag value 0x402 indicates that the request is a CMC request in binary format.

   - *Request*: The certificate request generated in step 2.

   The response is a PKCS #7 response (as specified in section 2.2.2.4).

   If the CA responds with an error (nonzero value) or the *Disposition* output parameter value (as specified in [MS-WCCE] section 3.1.1.6.2) is not "***Issued***" the HRA gets the next **CAToConnect** ADM element from the **remote CA name List** and performs step 4 again until a CA responds with a PKCS #7 response or all of the CAs in the **remote CA name List** have been tried.

5. If all of the CAs in the **remote CA name List** cannot be contacted or return an error (nonzero value), the HCEP Request processing must be aborted and HRA must respond with an error as specified in section 3.2.8.

- If the health certificate was obtained, then the HRA MUST do the following:

  - Set the HTTP message body of the HCEP Response to the ASN.1 DER encoded PKCS #7 response of the CA.

  - Set the **Content-Length** header of the HCEP Response message to the size of the ASN.1 DER encoded PKCS #7 response of the CA in bytes.

- If the health certificate was not obtained then the HRA MUST do the following:

  - Leave the HTTP message body of the HCEP Response empty.

  - Set the **Content-Length** header of the HCEP Response message to 0.

2. The HCEP response MUST then be sent to the client.

3.2.5.4   Creating Health Certificate Request by HRA

Create the Certificate Request using PKCS10 format

The health registration authority (HRA) MUST create a certificate request for use as a parameter for the interface **ICertRequestD::Request** as specified in [MS-WCCE] section 3.1.1.4.3.1.3 "New Certificate Request Using CMS and CMC Formats". This involves creating a certificate request using

PKCS#10 syntax and then encapsulating this request into a CMC certificate request message using CMS syntax. (PKCS#10 syntax is specified by [RFC2986], CMC protocol is specified by [RFC2797], and CMS syntax is specified by [RFC3852]). The certificate request in PKCS #10 syntax is created with the following fields:

- If the client is not authenticated, add a subject name field set to "CN=Unauthenticated System Health Authentication".

- Add the 1.3.6.1.4.1.311.2.1.14 (szOID_CERT_EXTENSIONS) attribute specified in [MS-WCCE] section 2.2.2.7.7 containing the following:

  - A Subject Alternative Name extension as specified in [RFC3280] section 4.2.1.7, with UPN and DNS names:

    - Add a DNS name (input) as Subject Alternative Name, obtained from the client certificate request.

    - Add a UPN (input) as Subject Alternative Name.

    - Add a SAN extension.

  - The HRA MAY add a Certificate Policies extension with these OIDs:<39>

    - A Certificate Policies extension as specified in [RFC3280] section 4.2.1.5, containing three instances of the PolicyInformation defined as follows:

      - If the client is compliant, set policyIdentifier to "1.3.6.1.4.1.311.47.1.10" (napPolicyInformationCompliantOid) specified in section 2.2.3.1 and no policyQualifiers. If the client is noncompliant, set policyIdentifier to "1.3.6.1.4.1.311.47.1.11" (napPolicyInformationNotCompliantOid) specified in section 2.2.3.2 and no policyQualifiers.

      - policyIdentifier set to "1.3.6.1.4.1.311.47.1.12" (napPolicyInformationIsolationStateOid) specified in section 2.2.3.3 and one instance of the PolicyQualifierInfo of type UserNotice, where explicitText is set to one of the following values:

        - Compliant.

        - Network connectivity is not being restricted but might be at a later time.

        - Noncompliant.

      - policyIdentifier set to "1.3.6.1.4.1.311.47.1.13" (napPolicyInformationExtendedStateOid) specified in section 2.2.3.4 and one instance of the PolicyQualifierInfo of type UserNotice, where explicitText is set to one of the following values:

        - No additional data.

        - Transition data.

        - Infected data.

        - Unknown data.

  - A key usage certificate extension as specified in [RFC3280] section 4.2.1.3, with the KeyUsage value set to 0x80 and marked as critical.

- A SKI extension as specified in [RFC3280] section 4.2.1.2.

- The HRA MAY add an EKU extension as specified in [RFC3280] section 4.2.1.13, with the following usages: 1.3.6.1.4.1.311.47.1.1 (napHealthyOid) specified in section 2.2.3.5 and 1.3.6.1.5.5.7.3.2 (szOID_PKIX_KP_CLIENT_AUTH).<40>

- The HRA MAY add Certificate Application Policy according to [MS-WCCE] section 2.2.2.7.7.3.<41>

  - When the client is compliant, add "1.3.6.1.4.1.311.47.1.1" (napHealthyOid); else add "1.3.6.1.4.1.311.47.1.3" (napUnhealthyOid) specified in section 2.2.3.6.

  - If the client is authenticated, add client auth application policy "1.3.6.1.5.5.7.3.2" (szOID_PKIX_KP_CLIENT_AUTH).

- The HRA MAY add Certificate Template OID Extension as specified in section [MS-WCCE] section 2.2.2.7.7.2.<42>

- The HRA MAY add other attributes for implementation-specific use to the request, as specified by PKCS #10 [RFC2986].

Create the CMC request and init from PKCS10

- Create a CMC request as specified in [MS-WCCE] section 3.1.1.4.3.1.3 and set TaggedRequest to the PKCS10 request created above.

- Add CMC Extensions: Add the addExtensions control attribute specified in [RFC2797] section 5.5 that contains the following certificate extensions:

  - Add application policy according to [MS-WCCE] section 2.2.2.7.7.3:

    - When the client is compliant, add "1.3.6.1.4.1.311.47.1.1" (napHealthyOid); else add "1.3.6.1.4.1.311.47.1.3" (napUnhealthyOid).

    - If the client is authenticated, add the client auth application policy "1.3.6.1.5.5.7.3.2" (szOID_PKIX_KP_CLIENT_AUTH).

  - Add the Extended Key Usage extension as specified in [RFC3280] section 4.2.1.13:

    - When the client is compliant, add "1.3.6.1.4.1.311.47.1.1" (napPolicyInformationCompliantOid); else add "1.3.6.1.4.1.311.47.1.3" (napPolicyInformationNotCompliantOid).

    - If the client is authenticated, add client auth application policy "1.3.6.1.5.5.7.3.2" (szOID_PKIX_KP_CLIENT_AUTH).

- If the HRA uses Enterprise CA, the HRA MAY add cert template extensions as specified in [MS-WCCE] section 2.2.2.7.7.1 and set the **Name** field that was configured by the administrator.<43>

- The HRA MAY add a control attribute 1.3.6.1.4.1.311.10.10.1 (szOID_CMC_ADD_ATTRIBUTES). The structure of the szOID_CMC_ADD_ATTRIBUTES type is identical to the Attributes type specified in [RFC2986] section 4.1. The structure SHOULD contain these attributes:<44>

  - The 1.3.6.1.4.1.311.13.2.3 (szOID_OS_VERSION) attribute as specified in [MS-WCCE] section 2.2.2.7.1.<45><46>

- The 1.3.6.1.4.1.311.21.20 (szOID_REQUEST_CLIENT_INFO) attribute as specified in [MS-WCCE] section 2.2.2.7.4.<47>

3.2.6 Timer Events

The server uses the following timer events:

- When the **CAConnectionDuration** timer specified in section 3.2.2 times out, an event is triggered. When this timer event is triggered, the HRA stops performing the processing steps for obtaining a health certificate from the CA specified in section 3.2.5.2 subsection 5, and responds with an error as specified in section 3.2.8.

- When the **SoHEvaluationDuration** timer specified in section 3.2.2 times out, the HCEP Request processing MUST be aborted and HRA MUST respond with an error as specified in section 3.2.8.

3.2.7 Other Local Events

None.

3.2.8 Error Handling

Error cases and typical error handling on the server MUST be as follows:

- Malformed HCEP request: If the server finds that the request is malformed according to the rules in section 3.2.5, it MUST respond with an HTTP Internal Server Error, as specified in [RFC2616] section 6.1.1.

- Failure to contact health policy server or certificate authority: If there is any error while trying to contact these servers, the HRA MUST respond with an HTTP Internal Server Error, as specified in [RFC2616] section 6.1.1.

4   Protocol Examples

The client determines through implementation-specific procedures that a health certificate is required. In a common scenario, the client is connected to a network, but the client does not have a valid health certificate for communicating on that network. To access the network, the client has to acquire a new health certificate. After the HCEP enrollment process is invoked, the following sequence of events occurs.

Compliant Client Example

1. The HCEA obtains the statement of health (SoH) from the system health entity.

2. The HCEA then generates a public-private key pair and constructs a health certificate request (see section 2.2.1.4).

3. The client creates the HCEP request (see section 2.2.1) and sends it to a preconfigured HRA URL.

4. The HRA receives the HCEP request, extracts the SoH from the certificate request, and passes the SoH on to a health policy server that evaluates the SoH in the request.

5. The health policy server responds with the statement of health response (SoHR), which contains the client's health state compliance to network policies.

6. If the client is compliant with network policies, the HRA obtains a health certificate for the certificate request in the HCEP request. This can be done by using the Windows Client Certificate Enrollment Protocol, as specified in [MS-WCCE].

7. The server creates an HCEP response (see section 2.2.2) and sends it to the client.

8. If a certificate was received in the HCEP response, the HCEA extracts the certificate and deposits it in the **PersistedComputerCertificates** ADM element specified in section 3.1.1.

Noncompliant Client Example

1. The HCEA obtains the SoH from the system health entity.

2. The HCEA generates a public-private key pair and constructs a health certificate request (see section 2.2.1.4).

3. The client creates the HCEP request (see section 2.2.1) and sends it to a preconfigured HRA URL.

4. The HRA receives the HCEP request, extracts the SoH from the certificate request and passes the SoH on to a health policy server that evaluates the SoH.

5. The health policy server responds with the SoHR and the noncompliant results of the evaluation of the client's health state compliance with network policies.

6. The HRA may obtain a health certificate for the certificate request in the HCEP request. This might be accomplished by using the Windows Client Certificate Enrollment Protocol, as specified in [MS-WCCE].

7. The server creates an HCEP response (see section 2.2.2) and sends it to the client.

8. If a certificate was received in the HCEP response, the HCEA extracts the certificate and deposits it in the **PersistedComputerCertificates** ADM element specified in section 3.1.1.

5   Security

5.1   Security Considerations for Implementers

The Health Certificate Enrollment Protocol does not ensure the authenticity of the statement of health (SoH) that is sent to the HRA. The implementation MUST use secure algorithms and methods to ensure the security of the SoH.

The health state of the machine sending the HCEP request can contain sensitive information. Therefore, implementers SHOULD ensure that the choice of transport for Health Certificate Enrollment Protocol messages is appropriate. For example, using HTTP over TLS (as specified in [RFC2818]) to authenticate the server and to provide confidentiality and integrity is better than using HTTP alone for Health Certificate Enrollment Protocol messages.

When the HCEA is authenticated by using Kerberos-based HTTP authentication (as specified in [RFC4559]), it allows the HRA to validate data that is present in the certificate request. HRA should not impersonate the client and should only identify the client using this authentication method. Otherwise, if the HRA is compromised, it can potentially allow attackers to impersonate clients.

5.2   Index of Security Parameters

Security parameter	Section
Transport protocol	2.1
Health certificate request	2.2.1.4
HRA authentication	2.1
HCEA authentication	1.5

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows Server 2008 operating system

- Windows Vista operating system

- Windows 7 operating system

- Windows Server 2008 R2 operating system

- Windows 8 operating system

- Windows Server 2012 operating system

- Windows 8.1 operating system

- Windows Server 2012 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 1.5: The Windows implementation of the HCEA depends on the administrator to configure the URLs of the HRAs. The administrator usually configures this manually or by using Group Policy (as defined in [MS-GPNAP]); in both cases, the HRA URL can be any HTTP URL compliant with RFC2616 section 3.2.2 or HTTPS URL as specified by RFC2818 section 2.4. As an alternative, the administrator can configure "HRA Automatic Discovery" as specified in [MSFT-HRA].

<2> Section 1.5: The Windows implementation of HCEA supports the RSA algorithm with a minimum key length of 1,024 bits. Key lengths longer than 2,048 bits are supported if the supporting cryptographic service providers (CSPs) are added to the system. For more information about the default list of CSPs that are present in a Windows-based system, see [MSDN-CSP].

<3> Section 1.5: The Windows implementations of the HCEA and the HRA depend on the **Cryptographic Application Programming Interface (CAPI)** (for more information, see [MSDN-CAPI]) for the implementation of the algorithms previously specified. The CAPI has a pluggable model for cryptographic service providers (CSPs) to be added, removed, or replaced by the administrator. Hence the HCEA and the HRA support all the algorithms that are provided by CAPI through the pluggable CSPs. For more information about the default list of CSPs that are present on a Windows-based system, see [MSDN-CSP].

<4> Section 1.5: Windows implementations of the HCEA and the HRA use the OID "1.3.6.1.4.1.311.47.1.1" (napHealthyOid) (as specified in section 1.9) for the **Statement of Health** field (as specified in section 2.2.1.4), the OID "1.3.6.1.4.1.311.47.1.3" (napUnhealthyOid) (as specified in section 2.2.2.4), and the OID "1.3.6.1.4.1.311.13.2.2" (szOID_ENROLLMENT_CSP_PROVIDER) (as specified in section 2.2.1.4).

<5> Section 1.6: The Windows HCEA depends on general platform security. Strong security is possible by using a secure execution environment.

<6> Section 2.2.1.1: The Windows implementation of the HCEA specifies the value of the **User-Agent** field as "NAP IPSec Enforcement v1.0".

<7> Section 2.2.1.1: The Windows implementation of the HRA ignores the user-agent field by default, but can be configured to accept HCEP requests containing only one of a predefined set of user agents.

<8> Section 2.2.1.2: The Windows implementation uses the same Correlation Id value as provided in SoH, as specified in [TNC-IF-TNCCSPBSoH] section 3.8.

<9> Section 2.2.1.4: The Windows implementation of HCEA follows the behavior recommended in this section when assigning a value to the subject token.

<10> Section 2.2.1.4: The Windows implementation uses the EKU OID value of "1.3.6.1.4.1.311.47.1.1" (napHealthyOid).

<11> Section 2.2.1.4: The Windows implementation uses the EKU OID value of "id-kp-clientAuth", as specified in [RFC3280] section 4.2.1.13, if the **ClientAuthenticationFlag** ADM element as specified in section 3.1.1 is set to TRUE.

<12> Section 2.2.1.4: The Windows implementation uses OID "1.3.6.1.4.1.311.13.2.2" (szOID_ENROLLMENT_CSP_PROVIDER) as a certificate extension in the certificate request.

<13> Section 2.2.2.2: The Windows implementation uses the same Correlation Id value that is provided in SoH, as specified in [TNC-IF-TNCCSPBSoH] section 3.8.

<14> Section 2.2.2.2: The values of the **HCEP-AFW-Protection-Level** and **HCEP-AFW-Zone** fields are set as metadata on the health certificates received from the HRA. Windows Vista, Windows 7, Windows 8, and Windows 8.1 have a component that behaves as an IPsec (for more information, see [MSFT-IPSEC]) policy source. The hint provided by the value of **HCEP-AFW-Zone** is used to select one of the preconfigured IPsec policies on the computer. The value of the **HCEP-AFW-Zone** field is ignored if it is not "1", "2", or "3".

<15> Section 2.2.2.2: The Windows implementation of the HRA does not implement any response header fields other than the fields mentioned in this document.

<16> Section 2.2.2.3: Regarding the HTTP message body in the HCEP response in the Windows implementation: If the HCEP client is compliant with health policies, HRA receives an SoHR from the health policy server, and HRA receives a certificate response (PKCS #7) from CA, then the HRA returns the certificate response in the HTTP message body. Otherwise, if the Health Certificate Enrollment Protocol client is not compliant with health policies, no message body is present in the HCEP response.

<17> Section 2.2.2.4: For Windows Vista SP1, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2:

Extended key usage: The health certificate response contains the OID "1.3.6.1.4.1.311.47.1.1" (napHealthyOid) if the client is compliant. If a server is configured to send a health certificate response when a client is noncompliant, the health certificate response contains the OID "1.3.6.1.4.1.311.47.1.3" (napUnhealthyOid).

MSApplicationPolicies: If present, this contains the following:

▪ The OID "1.3.6.1.4.1.311.47.1.10" (napPolicyInformationCompliantOid) if the client is compliant.

- The OID "1.3.6.1.4.1.311.47.1.11" (napPolicyInformationNotCompliantOid) if the client is noncompliant.

- The OID "1.3.6.1.4.1.311.47.1.12" (napPolicyInformationIsolationStateOid) for the compliance state with values:

  - 1: Compliant

  - 2: Network connectivity is not being restricted but may be at a later time

  - 3: Noncompliant

- The OID "1.3.6.1.4.1.311.47.1.13" (napPolicyInformationExtendedStateOid) for the extended state with values:

  - 0: No additional data

  - 1: Transition data

  - 2: Infected data

  - 3: Unknown data

Compliance and extended state are specified in [TNC-IF-TNCCSPBSoH].

<18> Section 3: Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2 can also act as "HCEP clients" to use HCEP to communicate with other servers running Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2.

<19> Section 3.1.1: The Windows implementation of the HCEA determines the value of the **ClientAuthenticationFlag** ADM element to be TRUE if the client computer is a part of an **Active Directory domain**; otherwise, the value of this element is FALSE.

<20> Section 3.1.1: In the Windows implementation, the **PersistedComputerCertificates** ADM element is part of the persisted local certificate storage of the autoenrollment feature that is described in [MS-CERSOD] section 2.1.2.2.2.

<21> Section 3.1.4: The Windows implementation of the HCEA re-enrolls for a health certificate when it determines a configuration change has occurred with respect to HRA URLs.

<22> Section 3.1.4: The Windows implementation has an empty list by default.

<23> Section 3.1.5.2: The Windows implementation of the HCEA does not validate the values of the HCEP header fields.

<24> Section 3.1.5.2: The Windows implementation of the HCEA does not compare the value of the HCEP-correlation-Id header to the value stored in the **CorrelationId** ADM element.

<25> Section 3.1.5.2: The Windows implementation of the HCEA does not discard the response if the HCEP-Correlation-Id in the HCEP response does not match the one that was sent in the corresponding HCEP request (as stored in the **CorrelationId** ADM element).

<26> Section 3.1.5.2: Windows implementations of HCEA use Certificate Store Functions (see [MSDN-CERTCERTSTR] for details) for storing certificates. Certificate Store Functions support associating metadata with the certificate by means of Certificate Context Properties; values of **AFW-Zone** and **AFW-Protection-Level** are stored in Custom Properties with IDs 0x08000 and 0x8001 respectively.

<27> Section 3.1.7: The Windows implementation of the HCEA sends an HCEP request to the configured servers when an IP address change is detected on any machine interface and when a previously unreachable request is made to the servers.

<28> Section 3.2.1: The Windows implementation allows any of these settings; however, by default, it does not contain any predefined server settings.

<29> Section 3.2.1: By default, the Windows implementation has an empty list. Example value: "NAP IPSec Enforcement v1.0".

<30> Section 3.2.1: By default, the Windows implementation has an empty list. Only OIDs are configured in the list. The name is provided in the following table for ease of readability.

Name	OID
RSA	1.2.840.113549.1.1.1
DSA	1.2.840.10040.4.1
DH	1.2.840.10046.2.1
RSASSA-PSS	1.2.840.113549.1.1.10
DSA	1.3.14.3.2.12
DH	1.2.840.113549.1.3.1
RSA_KEYX	1.3.14.3.2.22
mosaicKMandUpdSig	2.16.840.1.101.2.1.1.20
ESDH	1.2.840.113549.1.9.16.3.5
NO_SIGN	1.3.6.1.5.5.7.6.2
ECC	1.2.840.10045.2.1
ECDSA_P256	1.2.840.10045.3.1.7
ECDSA_P384	1.3.132.0.34
ECDSA_P521	1.3.132.0.35
RSAES_OAEP	1.2.840.113549.1.1.7
ECDH_STD_SHA1_KDF	1.3.133.16.840.63.0.2

<31> Section 3.2.1: By default, the Windows implementation has an empty list. Only OIDs are configured in the list. The name is provided in the following table for ease of readability.

Name	OID
sha1RSA	1.2.840.113549.1.1.5
md5RSA	1.2.840.113549.1.1.4
sha1DSA	1.2.840.10040.4.3
sha1RSA	1.3.14.3.2.29

Name	OID
shaRSA	1.3.14.3.2.15
md5RSA	1.3.14.3.2.3
md2RSA	1.2.840.113549.1.1.2
md4RSA	1.2.840.113549.1.1.3
md4RSA	1.3.14.3.2.2
md4RSA	1.3.14.3.2.4
md2RSA	1.3.14.7.2.3.1
sha1DSA	1.3.14.3.2.13
dsaSHA1	1.3.14.3.2.27
mosaicUpdatedSig	2.16.840.1.101.2.1.1.19
sha1NoSign	1.3.14.3.2.26
md5NoSign	1.2.840.113549.2.5
sha256NoSign	2.16.840.1.101.3.4.2.1
sha384NoSign	2.16.840.1.101.3.4.2.2
sha512NoSign	2.16.840.1.101.3.4.2.3
sha256RSA	1.2.840.113549.1.1.11
sha384RSA	1.2.840.113549.1.1.12
sha512RSA	1.2.840.113549.1.1.13
RSASSA-PSS	1.2.840.113549.1.1.10
sha1ECDSA	1.2.840.10045.4.1
sha256ECDSA	1.2.840.10045.4.3.2
sha384ECDSA	1.2.840.10045.4.3.3
sha512ECDSA	1.2.840.10045.4.3.4
specifiedECDSA	1.2.840.10045.4.3

<32> Section 3.2.1: By default, the Windows implementation has an empty list.

Example CSPs are as follows:

- Microsoft Base Cryptographic Provider v1.0

- Microsoft Base DSS and Diffie-Hellman Cryptographic Provider

- Microsoft Base DSS Cryptographic Provider

- Microsoft Base Smart Card Crypto Provider

- Microsoft DH SChannel Cryptographic Provider

- Microsoft Enhanced Cryptographic Provider v1.0

- Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider

- Microsoft Enhanced RSA and AES Cryptographic Provider

- Microsoft Exchange Cryptographic Provider v1.0

- Microsoft RSA SChannel Cryptographic Provider

- Microsoft Strong Cryptographic Provider

<33> Section 3.2.1: By default, the Windows implementation restricts the maximum size to 64 KB.

<34> Section 3.2.1: The value set to the **EvaluationTimeOut** ADM in the windows implementation of HRA is 500 milliseconds. This value cannot be changed or configured.

<35> Section 3.2.5.1: The Windows implementation of the HRA has the following behavior for the optional requirements that are listed in this section:

- Maximum size: by default, 64 KB. This can be configured to a different value.

- By default, accept all user agents. This behavior can be restricted to a configured list of user agents by specifying a substring of the user-agent values. Wildcards are not allowed in the list of user agents.

- By default, accept all public key algorithms. This behavior can be restricted to a configured list of public key algorithms.

- By default, accept all signature algorithms. This behavior can be restricted to a configured list of signature algorithms.

- By default, accept all CSPs. This behavior can be restricted to a configured list of CSPs.

<36> Section 3.2.5.3: This behavior is only available on a Server SKU implementation where HRA is installed, and was first released in Windows Server 2008.

<37> Section 3.2.5.3: The decision whether or not to request a health certificate for a presented SoH that is not compliant with the policy in the Windows implementation of the HRA is controlled by a setting in the HRA configuration located in the registry key: "Software\Microsoft\HCS\". The contents of the registry key are as follows:

Value: PolicyExtensionOn.

Type: REG_DWORD.

Data: A 32-bit unsigned integer.

Value	Meaning
0x00000000	HRA does not request health certificate for non-compliant SoH.
0x00000001	HRA requests health certificate for non-compliant SoH.

If the HRA is set to request a health certificate for a non-compliant SoH, the HRA communicates with the CA using the [MS-WCCE] protocol and the exact processing steps are described in section 2.2.2.3 of the Health Certificate Enrollment Protocol.

<38> Section 3.2.5.3: This behavior is only available on a Server SKU implementation where HRA is installed, and was first released in Windows Server 2008.

<39> Section 3.2.5.4: Certificate Policies are added when a certificate has to be issued for noncompliant clients.

<40> Section 3.2.5.4: Windows Server 2008 behavior only.

<41> Section 3.2.5.4: Windows Server 2008 behavior only.

<42> Section 3.2.5.4: Windows Server 2008 behavior only.

<43> Section 3.2.5.4: This behavior was first released in Windows Server 2008 R2.

<44> Section 3.2.5.4: The attribute is added to the CMC request in the Server SKU implementation where HRA is installed.

<45> Section 3.2.5.4: This behavior was first released in Windows Server 2008 R2 SKU.

<46> Section 3.2.5.4: The attribute is added to the CMC request in the Server SKU implementation where HRA is installed.

<47> Section 3.2.5.4: The attribute is added to the CMC request in the Server SKU implementation where HRA is installed.

7 Change Tracking

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

8 Index