Web Services Security Kerberos Token Profile 1.1

OASIS Standard incorporating Approved Errata, 01 November 2006

OASIS identifier:

`wss-v1.1-spec-errata-os-KerberosTokenProfile`

Location:

`http://docs.oasis-open.org/wss/v1.1`

Technical Committee:

Web Service Security (WSS)

Chairs:

Kelvin Lawrence, IBM
Chris Kaler, Microsoft

Editors:

Anthony Nadalin, IBM
Chris Kaler, Microsoft
Ronald Monzillo, Sun
Phillip Hallam-Baker, Verisign

Abstract:

This document describes how to use Kerberos [Kerb] tickets (specifically the AP-REQ packet) with the WSS: SOAP Message Security [WSS] specification.

Status:

This is an OASIS Standard document produced by the Web Services Security Technical Committee. It was approved by the OASIS membership on 1 February 2006. Check the current location noted above for possible errata to this document.

Technical Committee members should send comments on this specification to the technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at www.oasisopen.org/committees/wss.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Security Services TC web page (http://www.oasis-pen.org/who/intellectualproperty.shtml).

Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be vailable; neither does it represent that it has made any effort to identify any such rights. Information on

OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director. OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

This section is non-normative.

Table of Contents

1 Introduction

This specification describes the use of Kerberos [Kerb] tokens with respect to the WSS: SOAP Message Security specification [WSS].

Specifically, this document defines how to encode Kerberos tickets and attach them to SOAP messages. As well, it specifies how to add signatures and encryption to the SOAP message, in accordance with WSS: SOAP Message Security, which uses and references the Kerberos tokens.

For interoperability concerns, and for some security concerns, the specification is limited to using the `AP-REQ` packet (service ticket and authenticator) defined by Kerberos as the Kerberos token. This allows a service to authenticate the ticket and interoperate with existing Kerberos implementations.

It should be noted that how the `AP-REQ` is obtained is out of scope of this specification as are scenarios involving other ticket types and user-to-user interactions.

Note that Sections 2.1, 2.2, all of 3, and indicated parts of 6 are normative. All other sections are non-normative.

2 Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

2.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [2119].

Namespace URIs (of the general form "some-URI") represent some application-dependent or context-dependent URI as defined in RFC2396 [URI].

This specification is designed to work with the general SOAP [S11, S12] message structure and message processing model, and should be applicable to any version of SOAP. The current SOAP 1.2 namespace URI is used herein to provide detailed examples, but there is no intention to limit the applicability of this specification to a single version of SOAP.

2.2 Namespaces

The XML namespace [XML-ns] URIs that MUST be used by implementations of this specification are as follows (note that different elements in this specification are from different namespaces):

```
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd
http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd
```

Note that this specification does not introduce new schema elements.

The following namespaces are used in this document:

Prefix	Namespace
S11	`http://schemas.xmlsoap.org/soap/envelope/`
S12	`http://www.w3.org/2003/05/soap-envelope`

wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsse11	http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc#

The URLs provided for the `wsse` and `wsu` namespaces can be used to obtain the schema files. URI fragments defined in this specification are relative to the following base URI unless otherwise specified:

`http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1`

2.3 Terminology

Readers are presumed to be familiar with the terms in the Internet Security Glossary [ISG].

This specification employs the terminology defined in the WSS: SOAP Message Security Core Specification [WSS].

The following (non-normative) table defines additional acronyms and abbreviations for this document.

Term	**Definition**
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol
URI	Uniform Resource Identifier
XML	Extensible Markup Language

3 Usage

This section describes the profile (specific mechanisms and procedures) for the Kerberos binding of WSS: SOAP Message Security.

Identification: `http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1`

3.1 Processing Model

The processing model for WSS: SOAP Message Security with Kerberos tokens is no different from that of WSS: SOAP Message Security with other token formats as described in WSS: SOAP Message Security.

3.2 Attaching Security Tokens

Kerberos tokens are attached to SOAP messages using WSS: SOAP Message Security by using the `<wsse:BinarySecurityToken>` described in WSS: SOAP Message Security. When using this element, the `@ValueType` attribute MUST be specified. This specification defines six values for this attribute as defined in the table below:

URI	Description
`http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#Kerberosv5_AP_REQ`	Kerberos v5 AP-REQ as defined in the Kerberos specification. This `ValueType` is used when the ticket is an AP Request.
`http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ`	A GSS-API Kerberos V5 mechanism token containing an KRB_AP_REQ message as defined in RFC-1964 [1964], Sec. 1.1 and its successor RFC-4121 [4121], Sec. 4.1. This `ValueType` is used when the ticket is an AP Request (ST + Authenticator).
`http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#Kerberosv5_AP_REQ1510`	Kerberos v5 AP-REQ as defined in RFC1510. This `ValueType` is used when the ticket is an AP Request per RFC1510.

http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ1510	A GSS-API Kerberos V5 mechanism token containing an KRB_AP_REQ message as defined in RFC-1964, Sec. 1.1 and its successor RFC-4121, Sec. 4.1. This `ValueType` is used when the ticket is an AP Request (ST + Authenticator) per RFC1510.
http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#Kerberosv5_AP_REQ4120	Kerberos v5 AP-REQ as defined in RFC4120. This `ValueType` is used when the ticket is an AP Request per RFC4120
http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ4120	A GSS-API Kerberos V5 mechanism token containing an KRB_AP_REQ message as defined in RFC-1964, Sec. 1.1 and its successor RFC-4121, Sec. 4.1. This `ValueType` is used when the ticket is an AP Request (ST + Authenticator) per RFC4120.

It should be noted that the URIs in the table above also serve as the official URIs identifying the Kerberos tokens defined in this specification.

All token types defined in this section use the type 0x8003 defined in RFC1964 for the checksum field of the authenticator inside the AP_REQ.

The octet sequence of either the GSS-API framed KRB_AP_REQ token or an unwrapped AP_REQ is encoded using the indicated encoding (e.g. base 64) and the result is placed inside of the `<wsse:BinarySecurityToken>` element.

The following example illustrates a SOAP message with a Kerberos token.

```
<S11:Envelope xmlns:S11="..." xmlns:wsu="...">
    <S11:Header>
        <wsse:Security xmlns:wsse="...">
            <wsse:BinarySecurityToken EncodingType="http://docs.
               oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary" ValueType=" http://docs.oasis-
open.org/wss/oasis-wss-kerberos-token-profile-1.1#Kerb
erosv5_AP_REQ" wsu:Id="MyToken">boIBxDCCAcCgAwIBBaEDAgEOogcD...
            </wsse:BinarySecurityToken>
            ...
        </wsse:Security>
    </S11:Header>
    <S11:Body>
        ...
    </S11:Body>
</S11:Envelope>
```

3.3 Identifying and Referencing Kerberos Tokens

A Kerberos Token is referenced by means of the `<wsse:SecurityTokenReference>` element. This mechanism, defined in WSS: SOAP Message Security, provides different referencing mechanisms. The following list identifies the supported and unsupported mechanisms:

The `wsu:Id` MAY be specified on the `<wsse:BinarySecurityToken>` element allowing the token to be directly referenced.

A `<wsse:KeyIdentifier>` element MAY be used which specifies the identifier for the Kerberos ticket. This value is computed as the SHA1 of the pre-encoded octets that were used to form the contents of the `<wsse:BinarySecurityToken>` element. The `<wsse:KeyIdentifier>` element contains the encoded form the of the `KeyIdentifier` which is defined as the base64 encoding of the SHA1 result.

Key Name references MUST NOT be used.

When a Kerberos Token is referenced using `<wsse:SecurityTokenReference>` the `@wsse11:TokenType` attribute SHOULD be specified. If the `@wsse11:TokenType` is specified its value MUST be the URI that identifies the Kerberos token type as defined for a corresponding `BinarySecurityToken/@ValueType` attribute. The `Reference/@ValueType` attribute is not required. If specified, its value MUST be equivalent to that of the `@wsse11:TokenType` attribute..

The `<wsse:SecurityTokenReference>` element from which the reference is made contains the `<wsse:KeyIdentifier>` element. The `<wsse:KeyIdentifier>` element MUST have a `ValueType` attribute on the `<wsse:KeyIdentifier>` element with the value #Kerberosv5APREQSHA1 and its contents MUST be the SHA1 of GSS-API framed KRB_AP_REQ token or unwrapped AP-REQ, as appropriate, encoded as per the `<wsse:KeyIdentifier>` element's `EncodingType` attribute.

Reference Identifier	**ValueType URI**	Description
Kerberos v5 AP-REQ	`http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.1#Kerb`	SHA1 of the v5 AP-REQ octets, either GSS-API framed KRB_AP_REQ token or just the Kerberos AP-REQ.

	erosv5APREQSHA1	

The following example illustrates using ID references to a Kerberos token:

```
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="...">
    <S11:Header>
        <wsse:Security>
            <wsse:BinarySecurityToken EncodingType="http://docs.
oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/oasis-wss-
kerberos-token-profile-1.1#Kerberosv5_AP_REQ" wsu:Id="MyToken">
                  boIBxDCCAcCgAwIBBaEDAgEOogcD...
            </wsse:BinarySecurityToken>
            ...
               <wsse:SecurityTokenReference
TokenType="http://docs.oasis-open.org/wss/oasis-wss-kerberos-toke
n-profile-1.1#Kerberosv5_AP_REQ">
                   <wsse:Reference URI="#MyToken"
ValueType="http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-
profile-1.1#Kerberosv5_AP_REQ">
                   </wsse:Reference>
               </wsse:SecurityTokenReference>
            ...
        </wsse:Security>
    </S11:Header>
    <S11:Body>
        ...
    </S11:Body>
</S11:Envelope>
```

The AP-REQ packet is included in the initial message to the service, but need not be attached to subsequent messages exchanged between the involved parties. Consequently, the `KeyIdentifier` reference mechanism SHOULD be used on subsequent exchanges as illustrated in the example below:

```
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="...">
    <S11:Header>
        <wsse:Security>
            ...
               <wsse:SecurityTokenReference
wsse11:TokenType=http://docs.oasis-open.org/wss/oasis-wss-kerberos-
token-profile-1.1#Kerberosv5_AP_REQ>
```

```
                 <wsse:KeyIdentifier ValueType="http://docs.oasis-
open.org/wss/oasis-wss-kerberos-token-profile-1.1#Kerb
erosv5APREQSHA1">GbsDt+WmD9XlnUUWbY/nhBveW8I=
                 </wsse:KeyIdentifier>
               </wsse:SecurityTokenReference>
            ...
        </wsse:Security>
    </S11:Header>
    <S11:Body>
        ...
    </S11:Body>
</S11:Envelope>

```

3.4 Authentication

When a Kerberos ticket is referenced as a signature key, the signature algorithm [DSIG] MUST be a hashed message authentication code.

When a Kerberos ticket is referenced as an encryption key, the encryption algorithm MUST be a symmetric encryption algorithm.

The value of the signature or encryption key is constructed from the value of the Kerberos sub-key when it is present in the authenticator or a session key from the ticket if the sub-key is absent, either by using the Kerberos sub-key or session key directly or using a key derived from that key using a mechanism agreed to by the communicating parties.

3.5 Encryption

When a Kerberos ticket is referenced as an encryption key, the encryption algorithm MUST be a symmetric encryption algorithm.

The value of the signature or encryption key is constructed from the value of the Kerberos sub-key when it is present in the authenticator or a session key from the ticket if the sub-key is absent, either by using the Kerberos sub-key or session key directly or using a key derived from that key using a mechanism agreed to by the communicating parties..

3.6 Principal Name

Kerberos principal name definition and mapping of non-Kerberos names to Kerberos V principal names are out of scope of this document.

3.7 Error Codes

When using Kerberos tokens, it is RECOMMENDED to use the error codes defined in the WSS: SOAP Message Security specification. However, implementations MAY use custom errors, defined in private namespaces if they desire. Care should be taken not to introduce security vulnerabilities in the errors returned.

4 Threat Model and Countermeasures

The use of Kerberos assertion tokens with WSS: SOAP Message Security introduces no new message-level threats beyond those identified for Kerberos itself or by WSS: SOAP Message Security with other types of security tokens.

One potential threat is that of key re-use. The mechanisms described in WSS: SOAP Message Security can be used to prevent replay of the message; however, it is possible that for some service scopes, there are host security concerns of key hijacking within a Kerberos infrastructure. The use of the AP-REQ and its associated authenticator and sequencer mitigate this threat.

Message alteration and eavesdropping can be addressed by using the integrity and confidentiality mechanisms described in WSS: SOAP Message Security. Replay attacks can be addressed by using message timestamps and caching, as well as other application-specific tracking mechanisms. For Kerberos tokens ownership is verified by use of keys, so man-in-the-middle attacks are generally mitigated.

It is strongly recommended that GSS wrapped AP-REQ be used or that unwrapped AP-REQ be combined with timestamp be used to prevent replay attack.

It is strongly recommended that all relevant and immutable message data be signed to prevent replay attacks.

It should be noted that transport-level security MAY be used to protect the message and the security token in cases where neither a GSS-API framed KRB_AP_REQ token or an unwrapped AP-REQ combined with timestamp and signature are being used.

5 References

The following are normative references

[2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997

[Kerb] J. Kohl and C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt .

[KEYWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997

[S11] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

[S12] W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework", 23 June 2003.

[URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 3986, MIT/LCS, Day Software, Adobe Systems, January 2005.

[WSS] A. Nadalin et al., Web Services Security: SOAP Message Security 1.1 (WS-Security 2004), OASIS Standard, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.1.pdf.

[1964] J. Linn , The Kerberos Version 5 GSS-API Mechanism, RFC 1964, June 1996.

[4121] L, Zhu, K. Jaganathan, S. Hartman, The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2, RFC 4121, July 2005.

The following are non-normative references

[ISG] Informational RFC 2828, "Internet Security Glossary," May 2000.

[XML-ns] W3C Recommendation, "Namespaces in XML," 14 January 1999.

[DSIG] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, W3C Recommendation, 12 February 2002. http://www.w3.org/TR/xmldsig-core/.

Appendix A. Acknowledgments

Current Contributors:

Michael	Hu	Actional
Maneesh	Sahu	Actional
Duane	Nickull	Adobe Systems
Gene	Thurston	AmberPoint
Frank	Siebenlist	Argonne National Laboratory
Hal	Lockhart	BEA Systems
Denis	Pilipchuk	BEA Systems
Corinna	Witt	BEA Systems
Steve	Anderson	BMC Software
Rich	Levinson	Computer Associates
Thomas	DeMartini	ContentGuard
Merlin	Hughes	Cybertrust
Dale	Moberg	Cyclone Commerce
Rich	Salz	Datapower
Sam	Wei	EMC
Dana S.	Kaufman	Forum Systems
Toshihiro	Nishimura	Fujitsu
Kefeng	Chen	GeoTrust
Irving	Reid	Hewlett-Packard
Kojiro	Nakayama	Hitachi
Paula	Austel	IBM
Derek	Fu	IBM
Maryann	Hondo	IBM
Kelvin	Lawrence	IBM
Michael	McIntosh	IBM
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Bruce	Rich	IBM
Ron	Williams	IBM
Don	Flinn	Individual
Kate	Cherry	Lockheed Martin
Paul	Cotton	Microsoft
Vijay	Gajjala	Microsoft
Martin	Gudgin	Microsoft
Chris	Kaler	Microsoft
Frederick	Hirsch	Nokia
Abbie	Barbir	Nortel
Prateek	Mishra	Oracle
Vamsi	Motukuru	Oracle
Ramana	Turlapi	Oracle
Ben	Hammond	RSA Security

Rob	Philpott	RSA Security
Blake	Dournaee	Sarvega
Sundeep	Peechu	Sarvega
Coumara	Radja	Sarvega
Pete	Wenzel	SeeBeyond
Manveen	Kaur	Sun Microsystems
Ronald	Monzillo	Sun Microsystems
Jan	Alexander	Systinet
Symon	Chang	TIBCO Software
John	Weiland	US Navy
Hans	Granqvist	VeriSign
Phillip	Hallam-Baker	VeriSign
Hemma	Prafullchandra	VeriSign

Previous Contributors:

Peter	Dapkus	BEA
Guillermo	Lao	ContentGuard
TJ	Pannu	ContentGuard
Xin	Wang	ContentGuard
Shawn	Sharp	Cyclone Commerce
Ganesh	Vaideeswaran	Documentum
Tim	Moses	Entrust
Carolina	Canales-Valenzuela	Ericsson
Tom	Rutt	Fujitsu
Yutaka	Kudo	Hitachi
Jason	Rouault	HP
Bob	Blakley	IBM
Joel	Farrell	IBM
Satoshi	Hada	IBM
Hiroshi	Maruyama	IBM
David	Melgar	IBM
Kent	Tamura	IBM
Wayne	Vicknair	IBM
Phil	Griffin	Individual
Mark	Hayes	Individual
John	Hughes	Individual
Peter	Rostin	Individual
Davanum	Srinivas	Individual
Bob	Morgan	Individual/Internet2
Bob	Atkinson	Microsoft
Keith	Ballinger	Microsoft
Allen	Brown	Microsoft
Giovanni	Della-Libera	Microsoft
Alan	Geller	Microsoft
Johannes	Klein	Microsoft

Scott	Konersmann	Microsoft
Chris	Kurt	Microsoft
Brian	LaMacchia	Microsoft
Paul	Leach	Microsoft
John	Manferdelli	Microsoft
John	Shewchuk	Microsoft
Dan	Simon	Microsoft
Hervey	Wilson	Microsoft
Jeff	Hodges	Neustar
Senthil	Sengodan	Nokia
Lloyd	Burch	Novell
Ed	Reed	Novell
Charles	Knouse	Oblix
Vipin	Samar	Oracle
Jerry	Schwarz	Oracle
Eric	Gravengaard	Reactivity
Andrew	Nash	Reactivity
Stuart	King	Reed Elsevier
Martijn	de Boer	SAP
Jonathan	Tourzan	Sony
Yassir	Elley	Sun
Michael	Nguyen	The IDA of Singapore
Don	Adams	TIBCO
Morten	Jorgensen	Vordel

Appendix B. Revision History

Rev	Date	By Whom	What
errata	08-25-2006	Anthony Nadalin	Issue 456