SAML V2.0 Kerberos Web Browser SSO Profile Version 1.0

Committee Specification 01

07 February 2012

Specification URIs

This version:

http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/cs01/saml-kerberos-browser-sso-v1.0-cs01.odt (Authoritative)
http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/cs01/saml-kerberos-browser-sso-v1.0-cs01.html
http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/cs01/saml-kerberos-browser-sso-v1.0-cs01.pdf

Previous version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-kerberos-browser-sso-cd-01.odt (Authoritative)
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-kerberos-browser-sso-cd-01.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-kerberos-browser-sso-cd-01.pdf

Latest version:

http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/saml-kerberos-browser-sso-v1.0.odt (Authoritative)
http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/saml-kerberos-browser-sso-v1.0.html
http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/saml-kerberos-browser-sso-v1.0.pdf

Technical Committee:

OASIS Security Services (SAML) TC

Chairs:

Thomas Hardjono (hardjono@mit.edu), M.I.T.
Nathan Klingenstein (ndk@internet2.edu), Internet2

Editors:

Josh Howlett (josh.howlett@gmail.com), Individual
Thomas Hardjono (hardjono@mit.edu), M.I.T.
Nathan Klingenstein (ndk@internet2.edu), Internet2
Tom Scavo (tscavo@ncsa.uiuc.edu), National Center for Supercomputing Applications (NCSA)

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schema: http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/cs01/xsd/

Declared XML namespaces:

- urn:oasis:names:tc:SAML:2.0:profiles:kerberos:SSO:browser

Abstract:

The SAML V2.0 Kerberos Web Browser SSO Profile allows for transport of assertions using the Kerberos subject confirmation method by standard HTTP user agents with no modification of client software and maximum compatibility with existing deployments. The flow is similar to standard Web Browser SSO, but a Kerberos AP-REQ message is presented by the user agent via the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism. The presentation of a valid Kerberos AP-REQ message whose client principal name matches the principal name given in the subject confirmation strengthens the assurance of the resulting authentication context and protects against credential theft.

Status:

This document was last revised or approved by the OASIS Security Services (SAML) TC on the above date. The level of approval is also listed above.

Technical Committee members should send comments on this Work Product 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 http://www.oasis-open.org/committees/security/.

For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/security/ipr.php).

Citation format:

When referencing this Work Product the following citation format should be used:
[SAML2KrbSSO]
SAML V2.0 Kerberos Web Browser SSO Profile Version 1.0. 07 February 2012. OASIS Committee Specification 01.
http://docs.oasis-open.org/security/saml/Post2.0/saml-kerberos-browser-sso/v1.0/cs01/saml-kerberos-browser-sso-v1.0-cs01.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

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 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

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 available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on 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 implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

1 Introduction

In the scenario addressed by this profile, which is an alternate version of the SAML V2.0 Web Browser SSO Profile [SAML2Prof], a principal uses an HTTP user agent to access a web-based resource at a service provider. To do so, the user agent presents a SAML assertion that uses Kerberos subject confirmation [SAML2KSCM] acquired from its preferred identity provider.

The user may first acquire an authentication request from the service provider or a third party. The identity provider authenticates the principal by any method of its choosing and then produces a response containing at least one assertion using Kerberos subject confirmation and an authentication statement for the user agent to transport to the service provider. A Kerberos [RFC4120] AP-REQ message, supplied through the HTTP Negotiate authentication scheme [RFC4559] and the Kerberos GSS-API mechanism [RFC4121], proves the attesting entity's authorization to wield the Kerberos principal name bound to the assertion's Kerberos subject confirmation. Finally, the service provider consumes the assertion to create a security context for the principal.

In what follows, a profile of the SAML Authentication Request Protocol [SAML2Core] is used in conjunction with an HTTP binding (section 2.5). It is assumed that the user wields an HTTP user agent, such as a standard web browser, capable of presenting a Kerberos AP-REQ using with the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism.

1.1 Notation

This specification uses normative text. The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]:

> …they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)…

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

```
Listings of XML schemas appear like this.
```

```
Example code listings appear like this.
```

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
`md:`	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace defined in the SAML V2.0 metadata specification [SAML2Meta].
`ds:`	http://www.w3.org/2000/09/xmldsig#	This is the XML digital signature namespace defined in the XML Signature Syntax and Processing specification [XMLSig].
`hoksso:`	urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser	This is the web browser holder-of-key namespace defined by this document and its accompanying schema [HoKSSO-XSD].
`krbsso:`	urn:oasis:names:tc:SAML:2.0:profiles:kerberos:SSO:browser	This is the web browser Kerberos namespace defined by this document and its accompanying schema
`saml:`	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification

Prefix	XML Namespace	Comments
		[SAML2Core].
`samlp:`	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace defined in the SAML V2.0 core specification [SAML2Core].
`xs:`	http://www.w3.org/2001/XMLSchema	This is the XML Schema namespace [Schema1].

This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `Attribute`, **Datatype**, `OtherCode`.Terminology

1.2 Terminology

The term *Kerberos* as used in this specification refers to the Kerberos Network Authentication Service (V5) [RFC4120]. The terms *GSS* and *GSS-API* refer to the Generic Security Service Application Program Interface Version 2, Update 1 [RFC2743]. As used in this specification, these terms do not refer to any earlier versions of these protocols.

The term *TLS* as used in this specification refers to either the Secure Sockets Layer (SSL) Protocol 3.0 or any version of the Transport Layer Security (TLS) Protocol [RFC4346] [RFC5246]. As used in this specification, the term *TLS specifically does* ***not*** *refer to the SSL Protocol 2.0* [SSL2]*.*

1.3 Normative References

[HoKSSO-XSD] OASIS Committee Specification 01, Schema for SAML V2.0 Holder-of-Key Web Browser SSO Profile. July 2009. See http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.xsd

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

[RFC2246] T. Dierks, C. Allen. *The TLS Protocol Version 1.0*. IETF RFC 2246, January 1999. See http://www.ietf.org/rfc/rfc2246.txt

[RFC2743] J. Linn. Generic Security Service Application Program Interface Version 2, Update 1. IETF RFC 2743. See http://www.ietf.org/rfc/rfc2743.txt

[RFC4120] C. Neuman et al. *The Kerberos Network Authentication Service (V5)*. IETF RFC 4120, July 2005. See http://www.ietf.org/rfc/rfc4120.txt

[RFC4121] L. Zhu et al. *The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2*. IETF RFC 4121. See http://www.ietf.org/rfc/rfc4121.txt

[RFC4346] T. Dierks, E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. IETF RFC 4346, April 2006. See http://www.ietf.org/rfc/rfc4346.txt

[RFC4559] K. Jaganathan et al. *SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows*. IETF RFC 4559. See http://www.ietf.org/rfc/rfc4559.txt

[RFC5246] T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. IETF RFC 5246, August 2008. See http://www.ietf.org/rfc/rfc5246.txt

[SAML2Bind] OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[SAML2Core] OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAML2HoKAP] OASIS Committee Specification 01, *SAML V2.0 Holder-of-Key Assertion Profile*. July 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cs-01.pdf

[SAML2HoKWP] OASIS Committee Specification 01, *SAML V2.0 Holder-of-Key Web Browser SSO Profile*. July 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-01.pdf

[SAML2KSCM] OASIS Committee Draft 01, *SAML V2.0 Kerberos Subject Confirmation Method*. November 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-kerberos-subject-confirmation-method-cd-01.pdf

[SAML2Meta] OASIS Standard, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[SAML2Prof] OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[Schema1] H. S. Thompson et al. *XML Schema Part 1: Structures.* World Wide Web Consortium Recommendation, May 2001. See http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[SSL3] A. Freier, P. Karlton, P. Kocher. The SSL Protocol Version 3.0. Netscape Communications Corp., November 18, 1996. See http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt

[XMLSig] D. Eastlake, J. Reagle, D. Solo, F. Hirsch, T. Roessler. *XML Signature Syntax and Processing (Second Edition)*. World Wide Web Consortium Recommendation, 10 June 2008. See http://www.w3.org/TR/xmldsig-core/

1.4 Non-normative References

[IDPDisco] OASIS Committee Specification 01, *Identity Provider Discovery Service Protocol and Profile*., October 2007. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery-cs-01.pdf

[RFC4401] N. Williams. *A Pseudo-Random Function (PRF) API Extension for the Generic Security Service Application Program Interface (GSS-API) Mechanism*. IETF RFC 4401. See http://www.ietf.org/rfc/rfc4401.txt

[RFC4402] N. Williams. *A Pseudo-Random Function (PRF) for the Kerberos V Generic Security Service Application Program Interface (GSS-API) Mechanism*. IETF RFC 4402. See http://www.ietf.org/rfc/rfc4402.txt

[SAML2Secure] OASIS Standard, *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf

[SAML2Simple] OASIS Committee Draft 04, *SAMLv2.0 HTTP POST "SimpleSign" Binding*. December 2008. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-binding-simplesign-cd-04.pdf

[SSL2] K. Hickman. *The SSL Protocol*. Netscape Communications Corp., February 9, 1995. See http://www.mozilla.org/projects/security/pki/nss/ssl/draft02.html

2 Kerberos Web Browser Profile

2.1 Required Information

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:kerberos:SSO:browser`

Contact information: security-services-comment@lists.oasis-open.org

SAML Confirmation Method Identifiers: The SAML V2.0 Kerberos confirmation method identifier, `urn:oasis:names:tc:SAML:2.0:cm:kerberos`, is included in all assertions issued under this profile.

Description: Given below.

Updates: Provides an alternative to the SAML V2.0 Web Browser SSO Profile [SAML2Prof].

2.2 Background

This profile is designed to enhance the security of SAML assertion and message exchange without requiring modifications to client software. A SAML assertion using Kerberos subject confirmation [SAML2K-SCM] is delivered to the service provider via an HTTP binding (section 2.5). The user agent presents a Kerberos [RFC4120] AP-REQ message, resulting in a strong association of the resulting security context with the intended user and elimination of numerous attacks (section 4).

Enhanced security is the primary benefit associated with the use of this profile. Under ordinary Web Browser SSO, there is a small chance that a bearer token will be stolen in transit, as described in [SAML2Secure]. Confirming that the presenter of the token is the intended subject using the Kerberos protocol virtually eliminates this chance, improving the viability of SAML Web Browser SSO for sensitive applications.

2.3 Profile Overview

Figure 1 illustrates the basic template for achieving Web Browser SSO under this profile. The following steps are described by the profile. Within an individual step, there may be one or more actual message exchanges depending on the binding used for that step and other deployment-specific behavior.

1. **HTTP Request to Service Provider** (section 2.6.1)

   The principal, via an HTTP user agent, makes an HTTP request for a secured resource at the service provider. At this step, the user agent may or may not present a Kerberos [RFC4120] AP-REQ message to the service provider using the HTTP Negotiate authentication scheme [RFC4559] and the Kerberos GSS-API mechanism [RFC4121]. In any event, the service provider determines that no security context exists and subsequently initiates Kerberos Web Browser SSO.

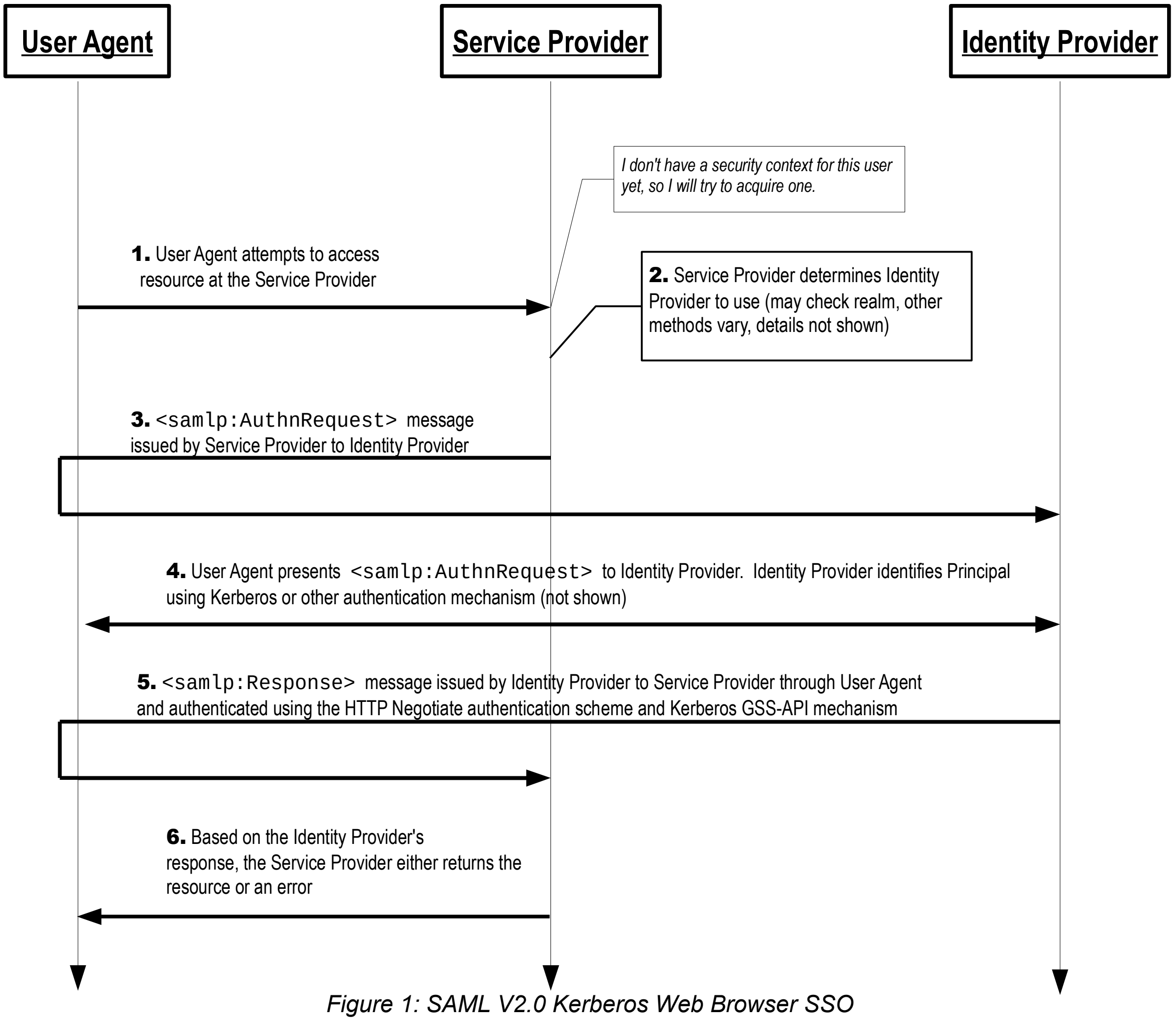
User Agent
Service Provider
Identity Provider
I don't have a security context for this user yet, so I will try to acquire one.
1. User Agent attempts to access resource at the Service Provider
2. Service Provider determines Identity Provider to use (may check realm, other methods vary, details not shown)
3. <samlp:AuthnRequest> message issued by Service Provider to Identity Provider
4. User Agent presents <samlp:AuthnRequest> to Identity Provider. Identity Provider identifies Principal using Kerberos or other authentication mechanism (not shown)
5. <samlp:Response> message issued by Identity Provider to Service Provider through User Agent and authenticated using the HTTP Negotiate authentication scheme and Kerberos GSS-API mechanism
6. Based on the Identity Provider's response, the Service Provider either returns the resource or an error

Figure 1: SAML V2.0 Kerberos Web Browser SSO

2. **Service Provider Determines Identity Provider** (section 2.6.2)

   The service provider determines the principal's preferred identity provider by any means.

3. **Service Provider Issues `<samlp:AuthnRequest>` to Identity Provider** (section 2.6.3)

   The service provider issues a `<samlp:AuthnRequest>` message to be delivered by the user agent to the identity provider. An HTTP binding is used (section 2.5) to transport the message to the identity provider through the user agent. The user agent presents the message to the identity provider in an HTTP request.

4. **Identity Provider Identifies Principal and Verifies Key Possession** (section 2.6.4)

   The principal is identified by the identity provider at this step. The identity provider identifies the principal using any authentication method at its disposal while honoring any requirements imposed by the service provider in the `<samlp:AuthnRequest>` message.

5. **Identity Provider Issues `<samlp:Response>` to Service Provider** (section 2.6.5)

   The identity provider issues a `<samlp:Response>` message to be delivered by the user agent to the service provider. The response either indicates an error or includes at least an authentication statement in an assertion using Kerberos subject confirmation. An HTTP binding is used (section 2.5) to transport the message to the service provider through the user agent. The user agent presents the message to the service provider in an HTTP request and, in conjunction with the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism, simultaneously presents a Kerberos AP-REQ message to the service provider as described in section 2.4.

6. **Service Provider Grants or Denies Access to Principal** (section 2.6.6)

   The SAML response is consumed by the service provider who either responds to the principal's user agent by establishing a security context for the principal and returning the requested resource, or by returning an error.

Note that an identity provider can initiate this profile at step 5 by issuing a `<samlp:Response>` message to a service provider without the preceding steps. The user agent or a third party may also initiate this profile by submitting an unsigned request at step 3.

2.4 Kerberos Usage

As noted in the introduction, this profile is an alternative to ordinary SAML Web Browser SSO [SAML2-Prof]. The primary difference between that profile and this Kerberos Web Browser SSO Profile is that the principal MUST present a Kerberos AP-REQ message whose client principal name matches the value given in the assertion's Kerberos subject confirmation. This leads to Kerberos subject confirmation, a type of subject confirmation that is stronger than the bearer subject confirmation inherent in ordinary Web Browser SSO.

The user agent presents a Kerberos AP-REQ message using the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism. The Kerberos AP-REQ message MUST satisfy the validity conditions required by the Kerberos protocol. This proves the user agent's authorization to present the assertion to the service provider.

According to the Kerberos protocol, authentication of the service is optional. Likewise this Kerberos Web Browser SSO Profile does not require Kerberos mutual authentication, which is strictly OPTIONAL. Moreover, the authentication method by which the identity provider identifies the principal is unspecified.

In summary, the principal MUST present a Kerberos AP-REQ message (using the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism) at step 5 (section 2.6.5). However, Kerberos authentication at step 1 (section 2.6.1) is strictly OPTIONAL. If the Kerberos protocol is not used to identify the user principal in step 1, the identity provider MUST be able to find a Kerberos user principal name that corresponds to the authenticated principal.

At either of steps 3 or 5 (or both), the identity provider or the service provider (resp.) MAY use the confirmed client principal name or the GSS context to create a security context for the principal. Also, at step 1, the service provider MAY use the confirmed client principal name or the GSS context to associate any subsequent exchange with the original request.

2.5 Choice of Binding

The identity provider and the service provider MUST use a browser-based HTTP binding to transmit the SAML protocol message to the other party. A SAML HTTP binding [SAML2Bind] MAY be used for this purpose:

1. HTTP Redirect
2. HTTP POST
3. HTTP Artifact

This profile does not preclude the use of other browser-based HTTP bindings (such as the SAML V2.0 SimpleSign binding [SAML2Simple]).

The identity provider and the service provider independently choose their preferred binding (subject to the other party's desire or ability to comply). The service provider chooses an HTTP binding to transmit the `<samlp:AuthnRequest>` message to the identity provider. Later, independent of the service provider's choice of binding, the identity provider chooses an HTTP binding to transmit the `<samlp:Response>` message to the service provider. The identity provider MUST NOT use the HTTP Redirect binding since the response typically exceeds the URL length permitted by most HTTP user agents.

If the service provider uses either the HTTP Redirect or HTTP POST binding, the `<samlp:AuthnRequest>` message is delivered directly to the identity provider at step 3 (section 2.6.3). If the service provider uses the HTTP Artifact binding, the identity provider uses the Artifact Resolution Profile [SAML2Prof] to make a callback to the service provider to retrieve the `<samlp:AuthnRequest>` message.

Similarly, if the identity provider uses the HTTP POST binding, the <samlp:Response> message is delivered directly to the service provider at step 5 (section 2.6.5). If the identity provider uses the HTTP Artifact binding, the service provider uses the Artifact Resolution Profile to make a callback to the identity provider to retrieve the <samlp:Response> message.

2.6 Profile Description

The SAML V2.0 Kerberos Web Browser SSO Profile is a profile of the SAML V2.0 Authentication Request Protocol [SAML2Core]. Where this Kerberos Web Browser SSO specification conflicts with Core, the former takes precedence.

If the request is initiated by the service provider, begin with section 2.6.1. If the request is initiated by the user agent or a third party, begin with section 2.6.4. If the identity provider issues a response without a corresponding request, begin with section 2.6.5. The descriptions refer to a single sign-on service and assertion consumer service in accordance with their use described in section 4.1.3 of [SAML2Prof]. Processing rules for all messages are specified in section 2.7 of this profile.

2.6.1 HTTP Request to Service Provider

The profile may be initiated by an arbitrary HTTP request to the service provider. The service provider is free to use any means it wishes to associate the subsequent interactions with the original request. For example, each of the SAML HTTP bindings discussed in section 2.5 provides a RelayState mechanism that the service provider MAY use to associate any subsequent exchange with the original request.

2.6.2 Service Provider Determines Identity Provider

The service provider determines the principal's preferred identity provider by any means at its disposal, including but not limited to the SAML V2.0 Identity Provider Discovery Profile [SAML2Prof] or the Identity Provider Discovery Service Protocol and Profile [IDPDisco].

If the user agent presented a Kerberos AP-REQ message at the previous step, the service provider may use the realm as a discovery hint by attempting to match this realm against values given in given in identity provider metadata (see section 2.8.1). If one or more candidate identity providers are found, the service provider MAY use this information to determine an appropriate identity provider.

2.6.3 Service Provider issues <samlp:AuthnRequest> to Identity Provider

Once an identity provider has been selected, the location of the single sign-on service to which to send a <samlp:AuthnRequest> message is determined based on the SAML binding chosen by the service provider (section 2.5). Metadata as described in section 2.8 MAY be used for this purpose. Following the HTTP request by the user agent in section 2.6.1, an HTTP response is returned containing a <samlp:AuthnRequest> message or an artifact, depending on the SAML binding used, to be delivered to the identity provider's single sign-on service.

Profile-specific rules for the contents of the <samlp:AuthnRequest> element are given in section 2.7.1.

2.6.4 Identity Provider Identifies Principal and Verifies Kerberos AP-REQ

The identity provider must perform two functions in this step: identify the principal presenting the <samlp:AuthnRequest> message and name this principal within a <saml:SubjectConfirmation> element.

The identity provider MUST establish the identity of the principal (or else it will return an error) prior to the issuance of the <samlp:Response> message. If the ForceAuthn attribute on the <samlp:AuthnRequest> element is present and true, the identity provider MUST freshly establish this identity rather than relying on any existing session it may have with the principal. Otherwise, and in all other respects, the identity provider may use any means to authenticate the user agent, subject to any requirements called out in the <samlp:AuthnRequest> message. In particular, the identity provider MAY use the Kerberos protocol to identify the principal, but this is by no means a requirement. See section 2.4 for details.

Any Kerberos <saml:SubjectConfirmation> elements included in the response MUST conform to the SAML V2.0 Kerberos Subject Confirmation Method. See section 2.7.3 for consequences of this dependency.

2.6.5 Identity Provider Issues <samlp:Response> to Service Provider

Depending on the SAML binding used (section 2.5), the identity provider returns an HTTP response to the user agent containing a <samlp:Response> message or an artifact, to be delivered to the service provider's assertion consumer service. Profile-specific rules regarding the contents of the <samlp:Response> element are included in section 2.7.3.

2.6.6 Service Provider Grants or Denies Access to Principal

As specified in section 2.4, the HTTP request that transports the response issued at the previous step MUST be authenticated using the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism. This supplies a Kerberos AP-REQ message naming a client principal to be matched against the Kerberos principal name bound to the assertion's <saml:SubjectConfirmation> element.

If the service provider is unable to authenticate the Kerberos AP-REQ message, the subject is not confirmed and the service provider SHOULD NOT create a security context for the principal.

Otherwise, the service provider MUST process the <samlp:Response> message and any enclosed <saml:Assertion> elements as described in [SAML2Core] and section 2.7.4 below. Any subsequent use of the <saml:Assertion> elements is at the discretion of the service provider and other relying parties, subject to any restrictions on use contained within the assertions themselves or previously established out-of-band policy governing interactions between the identity provider and the service provider.

To complete the profile, the service provider creates a security context for the user. The service provider MAY establish a security context with the user agent using any session mechanism it chooses. In particular, the Kerberos AP-REQ message MAY be used to create the security context as discussed in section 2.4.

2.7 Use of Authentication Request Protocol

This profile builds upon the Authentication Request Protocol [SAML2Core]. In the nomenclature of actors enumerated in section 3.4 of Core, the service provider is the request issuer and the relying party, the user agent is the attesting entity and the presenter, and the principal is the requested subject. There may be additional relying parties at the discretion of the identity provider.

2.7.1 <samlp:AuthnRequest> Usage

The use of the request MUST conform to section 2.7.1 of [SAML2HoKWP].

2.7.2 <samlp:AuthnRequest> Message Processing Rules

The processing of the request MUST conform to section 2.7.2 of [SAML2HoKWP].

2.7.3 <samlp:Response> Usage

If the identity provider wishes to return an error in response to a request, it MUST NOT include any assertions in the <samlp:Response> message. Otherwise, the <samlp:Response> element MUST conform to the following rules:

- The <saml:Issuer> element of the <samlp:Response> element MAY be omitted, but if present it MUST contain the unique identifier of the issuing identity provider. The Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- The response MUST contain at least one <saml:Assertion> element. Each assertion's <saml:Issuer> element MUST contain the unique identifier of the issuing identity provider, and

the Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

- The <saml:Subject> element of every assertion returned by the identity provider MUST refer to the authenticated principal using the SAML V2.0 Kerberos Subject Confirmation Method.
- Any <saml:Subject> elements in the response MUST strongly match the <saml:Subject> element in the <samlp:AuthnRequest> element (if any) as required by [SAML2Core]. If the <samlp:AuthnRequest> element contains an explicit <saml:SubjectConfirmation> element and the identity provider is unable to produce a strongly matching <saml:Subject> element for any reason, the identity provider MUST return an error.
- If the <samlp:AuthnRequest> element does not include a <saml:Subject> element, or the <saml:Subject> element in the request does not contain a <saml:SubjectConfirmation> element, every assertion in the response MUST use Kerberos subject confirmation as specified in [SAML2KSCM].
- Additional <saml:SubjectConfirmation> elements MAY be included in any assertion, though deployers should be aware of the implications of allowing weaker confirmation as the processing as defined in section 2.4.1.1 of [SAML2Core] is effectively satisfy-any. See section 3 for related considerations.
- Any assertion issued for consumption under this profile MUST contain a <saml:AudienceRestriction> element including the service provider's unique identifier in its <saml:Audience> element. Other conditions as defined in section 2.5 of [SAML2Core] (and other <saml:Audience> elements) MAY be included as requested by the service provider or at the discretion of the identity provider. All such conditions MUST be understood by and accepted by the service provider in order for the assertion to be considered valid.
- The set of one or more assertions MUST contain at least one <saml:AuthnStatement> element that reflects the authentication of the principal to the identity provider. Additional statements MAY be included in an assertion at the discretion of the identity provider.
- If the identity provider supports the Single Logout Profile [SAML2Prof], a <saml:AuthnStatement> element issued for consumption using this profile MUST include a SessionIndex attribute to enable per-session logout requests by the service provider.

As indicated above, the identity provider MUST issue at least one <saml:AuthnStatement> element. The identity provider typically issues exactly one such element but MAY issue multiple <saml:AuthnStatement> elements (in multiple assertions) if the service provider requires multiple assertions for various purposes.

If the identity provider issues multiple <saml:AuthnStatement> elements, the values of the IssueInstant attributes and the content of the <saml:SubjectLocality> elements MUST be identical across the <saml:AuthnStatement> elements. The content of the <saml:AuthnContext> elements MAY vary across the <saml:AuthnStatement> elements, presumably because the consumers of the various assertions have different requirements with respect to authentication context.

If the SAML HTTP POST binding (or a derivative of HTTP POST such as the SAML V2.0 SimpleSign binding [SAML2Simple]) is used to deliver the <samlp:Response> message to the service provider, every assertion in the response MUST be protected by digital signature. This can be accomplished either by signing each individual <saml:Assertion> element or by signing the <samlp:Response> element (or both).

2.7.4 <samlp:Response> Message Processing Rules

Regardless of the SAML binding used, the service provider MUST do the following:

- Verify any signatures present on the assertion(s) and/or the response.
- Verify that any assertions relied upon are valid according to processing rules in [SAML2Core].

- Using the Kerberos AP-REQ message presented by the user agent, any assertions using the Kerberos subject confirmation method in the response MUST be confirmed using the SAML V2.0 Kerberos Subject Confirmation Method.

Any assertion that is not valid, or whose subject confirmation requirements cannot be met, SHOULD be discarded and SHOULD NOT be used to establish a security context for the principal.

If the response contains multiple assertions with multiple `<saml:AuthnStatement>` elements, the service provider MAY consume any one of them at its discretion. How the service provider makes this decision is unspecified.

2.7.5 Artifact-Specific <samlp:Response> Message Processing Rules

If the HTTP Artifact binding (section 2.5) is used to deliver the `<samlp:Response>` message to the service provider, the dereferencing of the artifact using the Artifact Resolution Profile [SAML2Prof] MUST be mutually authenticated, integrity protected, and confidential. Mutually authenticated TLS or message signatures MAY be used to authenticate the parties and protect the messages.

The identity provider MUST ensure that only the service provider to whom the `<samlp:Response>` message has been issued is given the message as the result of a `<samlp:ArtifactResolve>` request. To partially satisfy this requirement, the identity provider MAY encrypt the assertions in the response.

2.8 Use of Metadata

This profile allows the use of two schema extensions to facilitate identity provider discovery and disambiguation of Web SSO profiles.

2.8.1 Identity Provider Discovery

This profile specifies an extension to the SAML V2.0 metadata specification [SAML2Meta] that allows the use of the `<krbsso:KerberosRealm>` element to name a Kerberos realm which an identity provider claims to authenticate. This element MUST be used as a child element of a role descriptor's `<Extensions>` element.

The following schema fragment defines the `<krbsso:KerberosRealm>` element :

```
<xs:element name="KerberosRealm" type="string"/>
```

2.8.2 Use of Bindings

Following the procedure specified in section 2.8 of [SAML2HoKWP], this profile specifies the use of the `Binding` attribute to disambiguate between this Kerberos Web Browser SSO Profile and other Web Browser SSO Profiles. The URI of the actual binding is instead placed into an extension attribute on the same endpoint element. The combined information is sufficient to distinguish the correct profile and binding when making a request to an endpoint.

All `<md:SingleSignOnService>` endpoints and all `<md:AssertionConsumerService>` endpoints to be used exclusively with this profile MUST have a `Binding` attribute of:

```
urn:oasis:names:tc:SAML:2.0:profiles:kerberos:SSO:browser
```

If an endpoint calls out the above `Binding` attribute value, it MUST also include the extension attribute `hoksso:ProtocolBinding` as described below. The XML attribute `hoksso:ProtocolBinding` contains the identifier of the desired protocol binding.

The following schema fragment defines the `hoksso:ProtocolBinding` attribute [HoKSSO-XSD]:

```
<xs:attribute name="ProtocolBinding" type="anyURI" use="optional"/>
```

3 Compatibility

Like the SAML V2.0 Web Browser SSO Profile [SAML2Prof], this Kerberos Web Browser SSO Profile is a profile of the SAML V2.0 Authentication Request Protocol [SAML2Core]. The primary difference between the original Web Browser SSO Profile and this Kerberos Web Browser SSO Profile is the mandate for Kerberos subject confirmation, made possible by the user agent's ability to present a Kerberos AP-REQ message using the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism. Although the SAML V2.0 Kerberos Web Browser SSO Profile is technically compatible with the original Web Browser SSO Profile, it is RECOMMENDED that separate endpoints be used to ensure all processing is performed in accordance with each profile's requirements and to avoid any negative impact on the user experience.

The SAML V2.0 Kerberos Web Browser SSO Profile does not preclude the addition of bearer `<saml:SubjectConfirmation>` elements in conforming assertions. This peculiar combination of `<saml:SubjectConfirmation>` elements is permitted since it is believed that carefully crafted deployments and use cases may find it useful. However, such hybrid assertions must be issued only after due deliberation and care. Technically, an assertion containing both bearer and Kerberos `<saml:SubjectConfirmation>` elements may be accepted as valid, reintroducing attacks such as man-in-the-middle and replay. Such assertions require security precautions appropriate for standard bearer assertions as described in section 7.1.1 of [SAML2Secure].

4 Security and Privacy Considerations

Assertions issued under the Kerberos Web Browser SSO Profile have different security and privacy characteristics than the bearer assertions used in the original Web Browser SSO Profile (see section 3). As specified, Kerberos subject confirmation minimizes the potential for assertion theft and virtually eliminates man-in-the-middle attacks. Potential replay attacks from unauthorized presenters that would otherwise require the tracking and checking of assertion `ID` attributes are also prevented by Kerberos subject confirmation.

4.1 Kerberos Usage

As a by-product of using Kerberos, as discussed in section 2.4, a security context resulting from an exchange conforming to the Kerberos Web Browser SSO Profile can be keyed using the client principal name that has been confirmed or the GSS context (for example, through the use of the Kerberos pseudo-random function extension for GSS-API [RFC4401] [RFC4402]). Application-layer sessions, such as those maintained by cookies, are often poorly protected by user agents, allowing for theft of the session and impersonation of the user. A session based on the client principal name or the GSS context has no such limitations, however.

4.1.1 Privacy Issues

In terms of privacy, there may be limitations on the degree to which users can remain anonymous under this profile since a Kerberos AP-REQ message contains a globally unique name for the subject, often containing personally identifying information.

4.2 Kerberos Client Authentication

The identity provider's requirements for user authentication as described in section 2.6.4 can be addressed by validating a presented Kerberos AP-REQ as described in [RFC4120]. This is not mandatory, however, unless such an authentication context is specifically requested by the service provider. Note that phishing is virtually eliminated in the presence of Kerberos client authentication, as there are greater challenges and no benefits to tricking the user into authenticating with a legitimate Kerberos AP-REQ message to a fraudulent party.

This profile offers potential usability benefits as well. If a Kerberos AP-REQ message is used for principal authentication, there is no need for the user to further confirm its identity, and potentially no user interaction is required.

5 Conformance

All parties, including the identity provider, the service provider, and the HTTP user agent, MUST conform to section 2.4. In particular, the user agent MUST have the ability to present a Kerberos AP-REQ message using the HTTP Negotiate authentication scheme and the Kerberos GSS-API mechanism.

The identity provider and the service provider MUST support the HTTP POST and HTTP Redirect bindings as discussed in section 2.5. Other binding support provided by the two parties is strictly OPTIONAL. In particular, support for the HTTP Artifact binding is OPTIONAL.

5.1 Identity Provider Conformance

In addition to the relevant requirements in section 5 above, an identity provider that conforms to this profile MUST adhere to the normative text in sections 2.6.4, 2.6.5, 2.7.2, and 2.7.3, and the relevant portions of section 2.7.5. If the identity provider uses SAML metadata, it MUST also conform to section 2.8 of this profile.

5.2 Service Provider Conformance

In addition to the relevant requirements in section 5 above, a service provider that conforms to this profile MUST adhere to the normative text in sections 2.6.1, 2.6.2, 2.6.3, 2.6.6, 2.7.1, and 2.7.4, and the relevant portions of section 2.7.5. If the service provider uses SAML metadata, it MUST also conform to section 2.8 of this profile.

Appendix A. Acknowledgments

The editor would like to acknowledge the contributions of the OASIS Security Services (SAML) Technical Committee, whose voting members at the time of publication were:

- John Bradley, Individual
- Scott Cantor, Internet2
- Duane DeCouteau, Veterans Health Administration
- Christian Guenther, Nokia Siemens Networks GmbH & Co.
- Frederick Hirsch, Nokia Corporation
- Ari Kermaier, Oracle Corporation
- Nathan Klingenstein, Internet2
- Hal Lockhart, Oracle Corporation
- Paul Madsen, NTT Corporation
- Kyle Meadors, Drummond Group Inc.
- Bob Morgan, Internet2
- Thinh Nguyenphu, Nokia Siemens Networks GmbH & Co.
- Rob Philpott, EMC Corporation
- Anil Saldhana, Red Hat
- Tom Scavo, National Center for Supercomputing Applications
- Kent Spaulding, Skyworth TTG Holdings Limited
- David Staggs, Veterans Health Administration
- Emily Xu, Sun Microsystems
- Phil Hunt (Oracle)
- Frederick Hirsch (Nokia)

The editor would also like to acknowledge the following particular individuals who contributed to the development of this document:

- Scott Cantor, Internet2
- Jeff Hodges, PayPal

Appendix B. Revision History

Document ID	Date	Committer	Comment
sstc-saml-kerberos-browser-sso-draft-01	Oct 16, 2009	J.Howlett	Initial draft
sstc-saml-kerberos-browser-sso-draft-02	8 Feb 2010	J.Howlett	Updated version of the draft
sstc-saml-kerberos-browser-sso-wd-02	11/21/11	T. Hardjono	Correction to normative reference to Kerberos XSD file.