SAML Implementation Profile

Deprecated

Note that this page has been deprecated. The information it contains is no longer current. See Kantara's SAML V2.0 Implementation Profile for Federation Interoperability for current information.

Community Review in progress!

This document contains **DRAFT** material intended for discussion and comment by the InCommon participant community. Comments and questions should be sent to the InCommon participants mailing list (participants@incommon.org).

DRAFT SAML Implementation Profile

This material is geared toward implementors of SAML software, primarily software that supports the SAML Web Browser SSO profile, though some of the material is generally applicable to other SAML profiles. It is not specifically aimed at deployers (i.e., the typical InCommon member).

This document summarizes a set of preliminary implementation requirements that may eventually form the basis of one or more InCommon SAML Implementation Profiles (hence the document title). Use of software supporting such a profile is not now, nor likely to be in the future, a requirement for participation in InCommon, but it does reflect the accumulation of experience around the software features required to support more robust and effective deployments within the federation, which elevates the federations value for all members.

These requirements are a superset of the implementation features required to support the saml2int deployment profile, which reflects a substantial amount of best practice within the global higher education community, and is itself expected to undergo further changes that will more closely align the deployment profile with the requirements outlined in this draft.

Because this document is a **DRAFT**, implementors are cautioned that it may (and likely will) change before taking final form and therefore implementors should be involved with the evolution of this document.

Contents

Basic Requirements

Extensibility

The extensibility of SAML is of paramount importance since it allows deployments to evolve and meet future needs. The SAML standard has explicit requirements for extensibility in both metadata and in protocol messages, and implementations MUST successfully consume any and all well-formed extensions. Most extension points in SAML have optional semantics, which means that ignoring extension content is a valid and acceptable practice. Unless otherwise noted as a required feature, `<Extensions>` elements in metadata and protocol messages MAY be ignored but MUST NOT result in software failures.

SAML Metadata

This section includes a basic set of SAML metadata requirements for both IdP and SP implementations. Support of metadata was strictly optional in the original SAML standard, but it is a critical component of modern SAML deployments.

1. MUST support SAML Metadata and the following OASIS specifications (linked on the OASIS Security Services Technical Committee wiki page):
   a. SAML V2.0 Metadata [SAML2MD] as updated by Errata [SAML2Errata]
   b. SAML V2.0 Metadata Schema [SAML2MD-xsd]
   c. SAML V2.0 Metadata Interoperability Profile [SAML2MDIOP]
2. MUST support the routine consumption of SAML metadata from a remote location via HTTP on a scheduled/recurring basis, with the content of the metadata automatically applied upon successful verification. HTTP caching and compression SHOULD be supported
3. MUST support the consumption of SAML metadata rooted in either `<md:EntityDescriptor>` or `<md:EntitiesDescriptor>` elements (in the latter case containing any number of child elements)

4. MUST support metadata verification based on verification of an XML Signature (see #algorithms for requirements) against a well-known key. Support for verification by certificate MAY be supported but it MUST be possible to configure verification based solely on the public key in a certificate.
5. MUST support metadata verification based on the presence of the `validUntil` XML attribute, and MUST have the ability to enforce limitations on the duration of validity (e.g., it must be possible to block consumption of metadata without such an attribute or one that is too far into the future)
6. Per [SAML2MDIOP], all run-time configuration of SAML profiles (technical trust and general operational configuration) MUST be manageable via SAML metadata alone. Further, it MUST be possible to configure an IdP or SP to allow basic interop with any peer for which metadata is supplied, without intervention by the deployer.
7. Also per [SAML2MDIOP], support for any number of long-lived, self-signed end entity certificates is REQUIRED, as is support for expired certificates, and certificates signed with any digest algorithm. Implementations MAY support alternative syntaxes for bare public keys with equivalent semantics to the same keys appearing in a certificate.
8. MUST support key rollover via SAML metadata (see #key-rollover requirements in the next section)

Key Rollover

To support seamless key rollover via SAML metadata, implementations MUST support the following features:

1. Implementations MUST be able to consume and utilize two or more signing keys bound to a single role descriptor in metadata. To verify a signature, an implementation MUST try each signing key (in unspecified order) until the signature is verified or there are no more signing keys (in which case signature verification fails).
2. Implementations MUST be able to consume and utilize two or more encryption keys bound to a single role descriptor in metadata. To encrypt a message, any encryption key in metadata MAY be used. If there are multiple encryption keys of a given type in metadata, the implementation may choose any one of them at its discretion and need not explicitly define which one will be used.
3. If an implementation supports inbound encryption, it MUST itself be configurable with up to two decryption keys (this is not a metadata requirement but applies to the configuration of keys used by the implementation).

> ⚠ **Default Key Descriptors in Metadata**
>
> An `<md:KeyDescriptor>` element in metadata that contains no `use` XML attribute MUST be valid as either a signing or encryption key.

Algorithms

Implementations MUST support the following XML Signature digest algorithms:

- http://www.w3.org/2001/04/xmlenc#sha256
- http://www.w3.org/2000/09/xmldsig#sha1

Implementations MUST support the following XML Signature signing algorithms:

- http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
- http://www.w3.org/2000/09/xmldsig#rsa-sha1

Implementations MUST support the following XML Encryption block encryption algorithms:

- http://www.w3.org/2001/04/xmlenc#aes128-cbc
- http://www.w3.org/2001/04/xmlenc#aes256-cbc
- http://www.w3.org/2009/xmlenc11#aes128-gcm
- http://www.w3.org/2009/xmlenc11#aes256-gcm

Implementations MUST support the following XML Encryption key transport algorithms:

- http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p
  - The following DigestMethod algorithms MUST be supported:
    - http://www.w3.org/2001/04/xmlenc#sha256
    - http://www.w3.org/2000/09/xmldsig#sha1
- http://www.w3.org/2009/xmlenc11#rsa-oaep
  - The following DigestMethod algorithms MUST be supported:
    - http://www.w3.org/2001/04/xmlenc#sha256
    - http://www.w3.org/2000/09/xmldsig#sha1
  - It is OPTIONAL to support mask generation functions other than http://www.w3.org/2009/xmlenc11#mgf1sha1

Support for other algorithms of all types is encouraged but not required.

IdP Requirements

All IdP implementations MUST satisfy the Basic Requirements listed above (#basic-requirements). In particular, a conforming IdP implementation MUST support the basic SAML Metadata requirements listed above (#saml-metadata).

Basic IdP Requirements

Implementations MUST support the *SAML V2.0 Web Browser SSO* profile as defined in SAML V2.0 Profiles [SAML2Prof] as updated by Errata [SAML2Errata] (linked on the OASIS Security Services Technical Committee wiki page). With respect to this profile:

1. MUST support the HTTP-Redirect and HTTP-POST bindings for requests
2. MUST support the HTTP-POST binding for responses
3. MUST support signing at either or both of the Response and Assertion layers
4. MUST support encryption of assertions
5. MUST support the MACE-Dir SAML Attribute Profiles
6. MUST support a “default attribute release policy” controlling the inclusion of SAML Attributes in an assertion such that if there is no policy associated with a requesting SP, then the default policy is used
7. MUST support the creation of attribute release policies tied to a specific SP by entityID
8. MUST support the “exact” `<RequestedAuthnContext>` operator (returning an error or selecting from among multiple login methods, as appropriate)
9. Assuming SP metadata is available, MUST respond to any valid AuthnRequest. If unable to satisfy a given request, must respond with a SAML error. (this requirement needs work, this is rooted in the general problem of SPs issuing requests to IDPs only to see them fall over or generate errors they don't get notice of)

In general, for the SAML profile(s) supported, all content and features defined by the standard should be evaluated for consideration, and if not supported should be fully documented as such.

Advanced IdP Requirements

1. MUST support dynamic (just-in-time) query of signed per-entity SAML metadata per the Metadata Query Protocol specifications. [MDQ-protocol] Verification of the XML Signature on the `<md:EntityDescriptor>` element is identical to the procedure outlined in the #saml-metadata section
2. MUST support the creation of attribute release policies that act based on the presence of specific `<mdattr:EntityAttributes>` extension elements [SAML2MDATTR] in SP metadata (via the 3-tuple Name, NameFormat, and AttributeValue)
3. MUST support dynamic customized login and consent pages based on the content of the `<mdui:UIInfo>` extension element [SAML2MDUI] in SP metadata
4. MUST support the creation of attribute release policies that act based on the presence of the `<mdrpi:RegistrationInfo>` extension element [SAML2MDRPI] in SP metadata (via the value of the `registrationAuthority` XML attribute)
5. MUST support the creation of attribute release policies that act based on the presence of specific `<md:RequestedAttribute>` elements in SP metadata (via the 3-tuple Name, NameFormat, and optionally AttributeValue(s))
6. MUST support the SAML 2 ForceAuthn feature
7. MUST support the SAML 2 IsPassive feature
8. MUST support all SAML-defined `<RequestedAuthnContext>` operators (exact, minimum, maximum, better)
9. MUST support the SAML V2.0 SingleLogout profile (in SAML2Core) and the SAML V2.0 Asynchronous Single Logout Protocol Extension [SAML2ASLO]
10. MUST support the use of the `<EncryptedID>` element, and in the case of decryption via at least two decryption keys simultaneously
11. MUST support the SAML V2.0 Enhanced Client or Proxy Profile Version 2.0 [SAML2ECP] with applicable requirements pertaining from the Browser profile requirements noted in the Basic requirements above
12. MUST support a mechanism for user-mediated consent to release attributes
13. MUST support the SAML V2.0 Metadata Profile for Algorithm Support [SAML2MDAlgSupport]

SP Requirements

> ⚠ The list of IdP requirements is a draft, but the list of SP requirements is **really** a draft. It's considerably less detailed and less reviewed at this stage.

All IdP implementations MUST satisfy the Basic Requirements listed above (#basic-requirements). In particular, a conforming SP implementation MUST support the basic SAML Metadata requirements listed above (#saml-metadata).

Basic SP Requirements

Implementations MUST support the *SAML V2.0 Web Browser SSO* profile as defined in SAML V2.0 Profiles [SAML2Prof] as updated by Errata [SAML2Errata] (linked on the OASIS Security Services Technical Committee wiki page). With respect to this profile:

1. MUST support the HTTP-Redirect and HTTP-POST bindings for requests
2. MUST support the HTTP-POST binding for responses
3. MUST support signing at either or both of the Response and Assertion layers
4. MUST support decryption of assertions via at least two decryption keys simultaneously
5. MUST support the processing of any string-based `<NameID>` format and any string-based `<saml:Attribute>` formulations, notably the MACE-Dir SAML Attribute Profiles
6. TBD: Require something related to IdP Discovery?

References

- [SAML2MDATTR] OASIS SAML V2.0 Metadata Extension for Entity Attributes Version 1.0 https://wiki.oasis-open.org/security/SAML2MetadataAttr (includes XML schema)
- [SAML2IDAssurance] OASIS SAML V2.0 Identity Assurance Profiles Version 1.0 https://wiki.oasis-open.org/security/SAML2IDAssuranceProfile
- [SAML2MDUI] OASIS SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0 https://wiki.oasis-open.org/security/SAML2MetadataUI (includes XML schema)
- [SAML2MD] Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0 http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf (includes XML schema)
- [SAML2MD-xsd] http://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd
- [SAML2MDIOP] OASIS SAML V2.0 Metadata Interoperability Profile Version 1.0 https://wiki.oasis-open.org/security/SAML2MetadataIOP
- [SAML2Errata] OASIS SAML V2.0 Errata 05 http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.pdf

- [SAML2Prof] Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
- [SAML2ASLO] OASIS SAML V2.0 Asynchronous Single Logout Protocol Extension Version 1.0 https://wiki.oasis-open.org/security/ASLO (includes XML schema)
- [SAML2MDAlgSupport] OASIS SAML V2.0 Metadata Profile for Algorithm Support Version 1.0 https://wiki.oasis-open.org/security/SAML2MetadataAlgSupport (includes XML schema)
- [SAML2ECP] OASIS SAML V2.0 Enhanced Client or Proxy Profile Version 2.0 https://wiki.oasis-open.org/security/SAML2EnhancedClientProfile (includes XML schema)
- [SAML2MDRPI] OASIS SAML V2.0 Metadata Extensions for Registration and Publication Information Version 1.0 https://wiki.oasis-open.org/security/SAML2MetadataDRI (includes XML schema)
- [SAML2SPReqInit] OASIS Service Provider Request Initiation Protocol and Profile Version 1.0 https://wiki.oasis-open.org/security/RequestInitProtProf (includes XML schema)
- [SAML2IdPDisco] OASIS Identity Provider Discovery Service Protocol and Profile https://wiki.oasis-open.org/security/IdpDiscoSvcProtonProfile (includes XML schema)
- [MDQ-protocol] Metadata Query Protocol Specifications (IETF) https://github.com/iay/md-query