SAMLDiffs

Differences Between SAML V2.0 and SAML V1.1

SAML V2.0 represents a significant feature upgrade to SAML V1.1. The enhancements include features derived from the Liberty Alliance Identity Federation Framework (ID-FF) V1.2 specifications that were contributed to the SSTC in 2003, capabilities present in the Internet2's Shibboleth architecture, and enhancement requests resulting from experience with numerous deployments of SAML V1.x in the industry.

The on-the-wire representations of SAML V2.0 assertions and protocol messages are incompatible with SAML V1.x processors. As is explained in the SAML Assertions and Protocols specification, only new major versions of SAML (of which this is one) typically cause this sort of incompatibility. In this release, much of the incompatibility is syntactic in nature; this was done for consistency and better component symmetry.

Specification Organization Changes

- The conformance specification now explicitly serves as the entry point for the SAML V2.0 OASIS Standard specifications.
- The assertion and protocol ("core") specification is now referred to as the Assertions and Protocols specification since it now defines multiple protocols.
- Processing rules are now clearly called out in each protocol.
- The single "bindings and profiles" specification has been split into two documents, one for bindings and one for profiles, and the latter now includes "SAML attribute profiles".
- There is a new authentication context specification and several accompanying XML schemas.
- There is a new metadata specification and an accompanying XML schema.
- Bibliographic references have been divided into normative and non-normative categories.
- There is a new non-normative executive overview document and a new technical overview document.

General Changes

- The SAML assertions namespace (known by its conventional prefix `saml:`) and protocols namespace (known by its conventional prefix `samlp:`) now contain the string `"2.0"` in recognition of this new major version of SAML.
- The `MajorVersion` and `MinorVersion` attributes that appeared on various elements have been combined into a single `Version` attribute that has the value `"2.0"`.
- The terminology used to describe various SAML system entities has been rationalized and enhanced to incorporate terminology from the Liberty Alliance. For example, SAML V2.0 adopts the term "identity provider" as opposed to "authentication authority".
- The SAML schema extensibility mechanisms have been rationalized and, in some cases, enhanced. XSD element substitution has been blocked in favor of type extension. The `<xs:anyAttribute>` wildcard has been added selectively to structures where it has been deemed valuable to add arbitrary "foreign" attributes without having to create a schema extension; these structures include subject confirmation data and SAML attributes.
- The authorization decision feature (statement and query) has been frozen; if more functionality is desired, it is recommended that XACML be used.
- A series of changes that were pre-announced during the SAML V1.x design cycles have been made:
- The deprecated `<AuthorityBinding>` element has been removed.
- The deprecated `<RespondWith>` element has been removed.
- The deprecated name identifier and artifact URI-based identifiers have been removed, to be replaced with other identifiers that were already introduced in SAML V1.1. Specifically:
  - Removed name identifier URI: `urn:oasis:names:tc:SAML:1.0:assertion#emailAddress`
    URI to use instead: `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`
  - Removed name identifier URI: `urn:oasis:names:tc:SAML:1.0:assertion#X509SubjectName`
    URI to use instead: `urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`
  - Removed name identifier URI: `urn:oasis:names:tc:SAML:1.0:assertion#WindowsDomainQualifiedName`
    URI to use instead: `urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName`
- Removed artifact URI: `urn:oasis:names:tc:SAML:1.0:cm:artifact-01`
  and also URI: `urn:oasis:names:tc:SAML:1.0:cm:artifact`
- URI references are now required to be absolute.
- The description of appearance of the `<Status>` element in SOAP messages has been improved.
- TBS: validity period semantics and syntax extended, removal of QNames in content, etc.

XML Signature and XML Encryption Support

- The `<ds:Signature>` element that allows for the digital signing of assertions and protocol messages has been positioned earlier in the respective content models.
- SAML now supports the use of the W3C XML Encryption recommendation to satisfy privacy requirements for several important SAML constructs.
- A new `<EncryptedID>` element has been defined that can hold an encrypted SAML identifier. These identifiers can be encrypted `<NameID>` or `<Assertion>` elements or elements of types derived from **NameIDType**, **AssertionType**, or **BaseIDAbstractType**.
- A new `<EncryptedAssertion>` element has been defined that can hold an encrypted SAML assertion.
- A new `<EncryptedAttribute>` element has been defined that can hold an encrypted SAML attribute.

Name Identifier, Subject, and Subject Confirmation Changes

- The new **BaseID** complex type is an extension point used to create new types of SAML identifiers.
- Name identifiers have new attributes permitting both IdP-specific and SP-specific qualification.
- New kerberos and entity name identifier formats

- Persistent and transient name identifier formats have been introduced that utilize pseudonyms to provide privacy-preserving characteristics for federated SAML identities.
- Restrictions on the use of the NameQualifier attribute
- The `<SubjectConfirmation>` element is now repeatable, with the formerly repeatable `<ConfirmationMethod>` element was renamed to `Method` and placed as an attribute within the `<SubjectConfirmation>` element.
- A set of generic attributes in `<SubjectConfirmationData>` have been defined for use in constraining the confirmation information. Overall assertion validity is more flexible within profiles as a result.
- A `<SubjectConfirmationData>` element now permits the inclusion of arbitrary XML attributes and child elements.
- A new **KeyInfoConfirmationDataType** complex type is used to constrain a `<SubjectConfirmationData>` element to hold `<ds:KeyInfo>` elements. Further, the usage of `<ds:KeyInfo>` within `<SubjectConformationData>` has been clarified to more clearly allow for impersonation.

General Assertion Changes

- The `AssertionID` attribute has been replaced by a general XML `ID` attribute.
- The `Issuer` attribute has been replaced by the `<Issuer>` element allowing the use of a generalized name identifier. By default, the value of the `<Issuer>` element is a URI of no more than 1024 characters.
- The `<Subject>` element has been moved up to be a child of the `<Assertion>` element rather than appearing as a child of a `<SubjectStatement>` element. All statements of the assertion must apply to the specified `<Subject>` element. The `<Subject>` element is now optional for extensibility reasons, although it is required for all assertions with SAML-specified statement types.
- The `<SubjectStatement>` element and its type have been removed.
- The `<Conditions>` element has been extended and restructured to permit more flexible conditions to be defined.
- The `<DoNotCacheCondition>` element has been replaced by a `<OneTimeUse>` element as a child of a `<Conditions>` element. The relationship of this condition to the `NotBefore` and `NotOnOrAfter` conditions has been delineated.
- A new `<ProxyRestriction>` element has been defined as a child of a `<Conditions>` element.

Authentication Statement Changes

- The `<AuthenticationStatement>` element has been renamed to `<AuthnStatement>`.
- The `<AuthnStatement>` element now supports the concept of a session in support of single logout and other session management requirements.
- The `AuthenticationMethod` attribute has been replaced by the new structured `<AuthnContext>` element permitting the expression of new, very fine-grained authentication methods and other authentication-related information.

Attribute Statement Changes

- The `<AttributeStatement>` element can now hold both encrypted and unencrypted SAML attributes.
- The name of the `AttributeName` field has been changed to just `Name`.
- The `AttributeNamespace` field has been removed in favor of `NameFormat`, and two new URI-based identifiers for attribute name format types have been defined for use in this field. This field can be left blank, as a default has been defined.
- Arbitrary XML attributes can now appear on the `<Attribute>` element without a supporting extension schema.
- Clearer instructions have been provided for how to represent null and multi-valued attributes.
- A series of attribute profiles has now been defined. They provide for proper interpretation of SAML attributes specified using common attribute /directory technologies.

General Request-Response Protocol Changes

- The `RequestID` and `ResponseID` attributes have been replaced by general XML `ID` attributes.
- The request datatype hierarchy has been reorganized; all queries are now kinds of requests, not inside requests, and the plain `<Query>` has been removed.
- `Consent` and `<Extensions>` constructs have been added to all requests and responses.
- An `<Issuer>` element can now be present on requests and responses (in addition to appearing on assertions).
- The response type hierarchy has been reorganized; most response elements in the various protocols are simply of **StatusResponseType**.
- New status codes have been added to reflect possible status values for the new protocols. Status codes are now URIs instead of Qnames.
- The `<AssertionIDRequest>` element is now used to obtain an assertion by means of its ID instead of using a `<Request>` with an `<AssertionIDReference>` element.
- SAML artifacts can no longer be used to refer to specific SAML assertions to be exchanged as described in the SAML V1.1 Browser/Artifact Profile. Artifacts are now used only to refer to SAML protocol messages. Once in possession of an artifact from a partner, an entity can retrieve the actual message from the partner through use of the new SAML Artifact Resolution Protocol. All types of protocol messages can theoretically be retrieved in this fashion.

Changes to SAML Queries

- An authentication query now supports the concept of sessions.
- In an authentication query, the `AuthenticationMethod` attribute has been replaced by the new structured `<AuthnContext>` element permitting queries for the new, very fine-grained authentication methods.
- In an attribute query, semantics have been defined to support the specification of attribute values as part of the query to limit the set of attribute values which may be returned.

New SAML Protocols

- The Authentication Request Protocol provides support for SP-initiated web SSO exchanges. This protocol allows the SP to make requests to an IdP and potentially control various aspects of the user authentication at the IdP, the binding to be used to return the response message, the set of SAML attributes to be included in the resulting assertion, etc. As part of this request, the SP can also indicate the desire to dynamically establish a new federated identity for the user.
- The Single Logout Protocol supports near-simultaneous logout of sessions at web SSO participants.
- The Artifact Resolution Protocol is used to retrieve SAML protocol messages through an artifact reference.
- The NameID Management Protocol provides the ability to modify federated name identifiers or to terminate their use.
- The NameID Mapping Protocol allows an SP that shares an identifier for a principal with an IdP to obtain a name identifier for the same principal in another format or that is in another federation namespace (i.e., is shared between the IdP and another SP).

Bindings Changes

- Generalized bindings have been created to support protocol message transfer between SAML parties using HTTP via a user agent (e.g., a browser). These bindings are known as the HTTP Redirect and the HTTP POST bindings.
- The HTTP Artifact Binding describes the means by which a SAML artifact can be transferred from one party to another. Once in possession of an artifact, an entity utilizes the SAML Artifact Resolution Protocol to retrieve the referenced protocol message.
- A PAOS (reverse SOAP) binding has been added.
- A set of mechanisms for relaying state have been added to most of the bindings.
- There is a new HTTP-based binding added for retrieval of assertions by means of URIs.

Profiles Changes

- A great deal of binding-specific detail has been factored out of the profiles. The resulting profiles are much shorter.
- The two original web browser profiles (Browser/Artifact and Browser/POST) have been consolidated into a single Web Browser SSO Profile.
- An enhanced client and proxy (ECP) SSO profile has been added.
- An Identity Provider Discovery Profile has been added that relies on the technique of creating common domain cookies.
- The new Artifact Resolution Profile describes how the Artifact Resolution Protocol is specifically used with the SOAP over HTTP Binding to retrieve SAML protocol messages referred to by an artifact.
- The new Name Identifier Mapping Profile describes how the Name Identifier Mapping Protocol is specifically used with the SOAP over HTTP Binding.
- As noted earlier, a series of attribute profiles has now been defined.