SAML V2.0 Metadata Guide

Editor:
Rainer Hörbe

Abstract:
The guide provides an overview of the SAML metadata specification, with a focus on frequently used elements and attributes. The target audience are developers and administrators who want to become familiar with SAML Metadata before going into details with the specification in various normative specifications and XML schema documents.

Contents
1 Types............................................................................................................................................2
  1.1 Organization............................................................................................................................2
  1.2 KeyDescriptor........................................................................................................................2
  1.3 IDPSSODescriptor....................................................................................................................2
  1.4 SSODescriptorType................................................................................................................3
  1.5 SPSSODescriptor....................................................................................................................3
  1.6 RoleDescriptor.......................................................................................................................3
  1.7 EntityDescriptor.....................................................................................................................4
  1.8 AttributeConsumingService ................................................................................................4
  1.9 AttributeAuthorityDescriptor...............................................................................................4
  1.10 AffiliationDescriptor............................................................................................................4
  1.11 EntitiesDescriptor................................................................................................................4
  1.12 EndpointType.......................................................................................................................5
  1.13 ContactType........................................................................................................................5
2 Elements.....................................................................................................................................5
3 Relationships..............................................................................................................................6
4 Metadata Specification Cross Reference....................................................................................7
1 Introduction

SAML metadata is configuration data required to automatically negotiate agreements between system entities, comprising identifiers, binding support and endpoints, certificates, keys, cryptographic capabilities and security and privacy policies. Several SAML specifications standardize metadata structures and contents, thus supporting efficient management processes for scaleable deployments.

As SAML specifications evolved over time metadata related specifications are contained in several documents. This non-normative document provides a consolidated overview and links to the normative specifications.

1.1 Metadata Use Cases

Identity management systems, including identity federations, need to have a common understanding about which entities are part of the system, and the configuration parameters of the participating system entities. This configuration is called metadata and can be exchanged in an informal and unstructured way, or standardized and structured with SAML metadata. The following diagrams show both approaches:

(A) System entities exchanging configuration data in a bidirectional and ad-hoc manner
(B) A federation operator (FO) manages metadata and provides metadata feed

Data flows in model (B) with coordinated metadata exchange follow this pattern:
A typical scenario has following use cases for metadata management:

System entity create self-describing metadata and consume metadata for other system entities. A federation operator is managing the processes to validate, aggregate and publish metadata in an assured manner. These use cases provide the bases for other use cases, such as adding a system entity, establishing and revoking technical trust, key rollover, attribute release and communicating an entity's conformance to some policy.

SAML Metadata specifications enable that processes exchange data required for those use cases in an interoperable way.

1.2 Metadata by Example

The key building block for SAML metadata is the EntityDescriptor, which describes a system entity such as an Identity Provider or Service Provider.

The EntitiesDescriptor container element is a common method to aggregate and sign a set of EntityDescriptor elements. Its validUntil and cacheDuration attributes deliver the data for keeping the metadata fresh.

This EntityDescriptor contains the role(s) (IdP in this case) and contact information.

The IDPSSODescriptor comprises 4 areas of data:

- Algorithm support: announcing multiple secure algorithms provides some flexibility and reduces the time to phase out a broken cipher;
- Discovery support: This elements are used to provision an IdP discovery service;
- Signing/encryption keys: As alternative to an PKIX infrastructure entities may do key validation and decryption based on metadata; Signing keys can be used for TLS endpoint validation of SOAP endpoints as well;
- Endpoints and nameIDs: URLs and bindings for a single endpoint in this example are supplied in addition the the supported NameIDFormat.

An metadata document with one IdP and one SP might look like this:
The SPSSODescriptor is similar to the IdP, however must have an `<AssertionConsumerService>` but no `<SingleSignOnService>`. In addition if a federation has many SPs `EntityCategories` might be useful to better manage attribute release.

Metadata example with one IdP and SP each

The SPSSODescriptor is similar to the IdP, however must have an `<AssertionConsumerService>` but no `<SingleSignOnService>`. In addition if a federation has many SPs `EntityCategories` might be useful to better manage attribute release.
2 Elements

As SAML Metadata is an XML-based specification, data structures are build from elements and attributes arranged in a mostly hierarchical style.

2.1 AttributeAuthorityDescriptor

The use of the <AttributeAuthorityDescriptor> role is generally a compatibility requirement for supporting legacy or other SPs that rely on queries for attributes. Another use case is Backend Attribute Exchange, where users authenticate directly to a SP, which will obtain attributes from the Attribute Authority. In most cases, much of the role content will be identical with that from <IDPSSODescriptor>.

The order of all this information is significant, which you can refer to the schema for, but the most common elements included would be present in the following order:

- <KeyDescriptor> (can be omitted, but rarely)
- <AttributeService> (always at least one)
- <NameIDFormat> (if any)
- <Attribute> (rare today, but may be reasonable to include)

Reference: [SAML2Meta]

2.2 AffiliationDescriptor

While the <EntityDescriptor> container element primarily describes a specific system entity, an entity with the affiliation role might alternatively represent a group of other entities, such as related service providers that share persistent NameIDs.

Reference: [SAML2Meta]

2.3 AttributeConsumingService

SPs support SSO protocols by including one or more <AssertionConsumerService> endpoint elements in their metadata. These are the locations to which the IdP will eventually send the user at the SP. By enumerating them in the metadata, the IdP can ensure that the user's information is sent only to authorized locations. However, as attribute release is frequently restricted by privacy considerations, the use of entity categories (see below) may be a more appropriate way than listing requested attributes in the SP's EntityDescriptor.

For technical reasons, these endpoint elements have to carry an additional index XML attribute, which should generally contain a small positive integer. The index values should be unique across all the like-named elements within the role.

Metadata can be used to advertise "service offerings" that can be referenced for service discovery, possibly grouped with an IdP-initiated authentication. An SP can define multiple services or service levels, with accompanying human-readable descriptions, to drive the development of IdP-provided service discovery user interfaces.

Reference: [SAML2Meta]

2.4 ContactPerson, Organization

Below the <EntityDescriptor> (or <RoleDescriptor>) level, there are optional <ContactPerson> and <Organization> elements that provide information about who is standing up a particular entity and how to contact them.
For testing purposes, one will rarely if ever need to supply these elements, but they may be
needed for production use. Organization metadata in particular often gets used by other
software systems that consume metadata in order to present lists of entities with human-
readable names. Examples of such systems include IdPDiscovery services or software to
assist users in granting consent for login and release of attributes to SPs.

Reference: [SAML2Meta]

2.5 Endpoint

The EndpointType describes a protocol binding endpoint at which a SAML entity can be sent
protocol messages. Various protocol or profile-specific metadata elements are bound to this
instances of this type, using the Binding, Location and ResponseLocation attributes.

Examples for specific instances are
  • <SingleSignOnService>
  • <SingleLogoutService>
  • <AttributeService>

Reference: [SAML2Meta]

2.6 EntityAttribute

The <EntityAttributes> element provides a generic extension mechanism for carrying an arbitrary set
of attribute information for an <EntityDescriptor> or a group thereof. It is typically used for entity
categories.

Reference: [SAML2MDext-EA]

2.7 EntityCategory

An EntityCategory assigns category membership semantics to an entity. Categories have been
defined for SPs that conform to a data protection Code of Conduct, or are members of a group that is
eligible to receive a certain set of user attributes.

Reference: [MacedirEntityCat]

2.8 EntitiesDescriptor

<EntitiesDescriptor> is a container, which usually wraps one or more elements of the type
<EntityDescriptor>. It enables a set of IdPs or SPs to be described at once, and then signed as
a unit. This is a common way for federations to supply metadata about their members. It is
also frequently used in deployments where metadata has to be maintained more or less by
hand, because it allows a bunch of information to be maintained in one file, with the entire set
of metadata supplied to an IdP or SP with a single configuration element. It's usually easier
that way than to individually supply metadata about a single entity at a time using multiple
metadata sources.

Reference: [SAML2Meta]

2.9 EntityDescriptor

An <EntityDescriptor> represents a system entity in metadata, which is a SAML service, such
as an IdP or an SP. Each entity has an entityID, that is a unique name that distinguishes it from
any other entity.
The value of the entityID attribute SHOULD be the canonical URL of the entity's metadata document. Canonical URLs follow the semantic-preserving normalization as specified in RFC 3986 section 6. The entity ID should accurately reflects the organization that owns the entity.

Reference: [SAML2Meta]

### 2.10 IDPSSODescriptor

An IdP role typically includes the following descriptive information:

- the public key(s) used by the IdP for authentication and encryption
- endpoints of various types for communicating with it
- explicitly supported identifier formats, if any
- explicitly supported attributes, if any

The order of all this information is significant, which you can refer to the schema for, but the most common elements included would be present in the following order:

- `<Extension>` (optional IdP discovery and algorithm support)
- `<KeyDescriptor>` (can be omitted, but rarely)
- `<ArtifactResolutionService>` (only needed if supporting response by artifact)
- `<SingleLogoutService>` (if any)
- `<NameIDFormat>` (if any)
- `<SingleSignOnService>` (always at least one)

Reference: [SAML2Meta]

### 2.11 KeyDescriptor

A common use of Metadata across both IdP and SP roles, among others, is to associate one or more public keys with the system being defined. The `<KeyDescriptor>` element is a wrapper around the XML Signature-defined `<ds:KeyInfo>` element, an extensible container for describing keys.

Because this container is so extensible, there is no prescribed behavior for the use of this element within metadata. Broadly speaking, there are two general approaches: inline and indirect.

**Inline Keys**

The approach that has been standardized as part of [SAML2MetaIOP] at OASIS is to directly express an explicitly trusted public key using either a `<ds:KeyValue>` or `<ds:X509Certificate>` element. Any number of keys can be included to manage key rollover, and individual keys can be labeled for authentication, encryption, or both.

A full description of the security implications of metadata usage is beyond the scope of this material, but be advised that this profile generally requires the enclosing metadata include an expiration time using a validUntil XML attribute. This prevents older signed metadata containing retired or compromised keys from being accepted.

**Indirect Key References**

The indirect approach involves describing a public key for use as input to a separately-controlled trust evaluation process. This is common to commercial SAML implementations, and may include a wide range of approaches to representing a key, including key "names" using `<ds:KeyName>` or `<ds:X509Subject>` elements, certificate references using the `<ds:X509IssuerSerial>` element, or an actual certificate that is subjected to additional validation using other rules defined outside of metadata.

Reference: [SAML2Meta], [SAML2MetaIOP]
2.12 Organization
Optional element identifying the organization responsible for the SAML entity described by the element.
Reference: [SAML2Meta]

2.13 Roles
Below the <EntityDescriptor>, the main information unit is the "role". An <EntityDescriptor> contains one or more <RoleDescriptors>, which is a super-type for roles such as <IDPSSODescriptor> and <AttributeAuthorityDescriptor> in the case of metadata about IdPs and <SPSSODescriptor> in the case of metadata about SPs.

One important piece of information common to all role elements is the protocolSupportEnumeration attribute, which MUST be present. This attribute contains a space-delimited collection of URIs that represent general classes of protocol support for the role in question. There are URIs defined by the various standards and profiles to represent the fact that an entity acting in a role "supports" a particular protocol family, such as SAML 2.0.
Reference: [SAML2Meta]

2.14 SPSSODescriptor
An SP role typically includes the following descriptive information:

- the public key(s) used by the SP for authentication and encryption
- endpoints of various types for communicating with it
- explicitly supported identifier formats, if any
- descriptions of the "services" offered by the SP and the SAML attributes required by them, possibly abstracted into a entity category.

The order of all this information is significant, which you can refer to the schema for, but the most common elements included would be present in the following order:

- <Extension> (optional service discovery, algorithm support and entity category)
- <KeyDescriptor> (can be omitted, but rarely)
- <SingleLogoutService> (if any)
- <NameIDFormat> (if any)
- <AssertionConsumerService> (always at least one)
- <AttributeConsumingService> (rare today, but good practice to include)

Reference: [SAML2Meta]
3 Relationships

The diagram below gives an abridged view on the relationship between the most important SAMI metadata elements in UML syntax.
## 4 Metadata Specification Cross Reference

This list shows documents that define metadata structures and the corresponding schema file name.

<table>
<thead>
<tr>
<th>Short</th>
<th>Title</th>
<th>Schema File</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML2Meta</td>
<td>Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0</td>
<td>saml-schema-metadata-2.0.xsd</td>
</tr>
<tr>
<td>SAML2MetaIOP</td>
<td>SAML V2.0 Metadata Interoperability Profile Version 1.0</td>
<td></td>
</tr>
<tr>
<td>SAML2MDext-QR</td>
<td>Metadata Extension for SAML V2.0 and 3 V1.x Query Requesters</td>
<td>sstc-saml-metadata-ext-query.xsd</td>
</tr>
<tr>
<td>IdpDisco</td>
<td>Identity Provider Discovery Service Protocol and Profile</td>
<td>sstc-saml-idp-discovery.xsd</td>
</tr>
<tr>
<td>SAML2MDext-EA</td>
<td>SAML V2.0 Metadata Extension for Entity Attributes</td>
<td>sstc-metadata-attr.xsd</td>
</tr>
<tr>
<td>SAML2Assur</td>
<td>SAML V2.0 Identity Assurance Profiles</td>
<td></td>
</tr>
<tr>
<td>SAML2ReqInit</td>
<td>Service Provider Request Initiation Protocol and Profile</td>
<td>sstc-request-initiation.xsd</td>
</tr>
<tr>
<td>SAML2MetaAlgSup</td>
<td>SAML v2.0 Metadata Profile for Algorithm Support</td>
<td>sstc-saml-metadata-algsupport-v1.0.xsd</td>
</tr>
<tr>
<td>SAML2MetaRPI</td>
<td>SAML V2.0 Metadata Extensions for Registration and Publication Information</td>
<td>saml-metadata-rpi-v1.0.xsd</td>
</tr>
<tr>
<td>SAML2MetaUI</td>
<td>SAML V2.0 Metadata Extensions for Login and Discovery User Interface</td>
<td>sstc-saml-metadata-ui-v1.0.xsd</td>
</tr>
<tr>
<td>SAML2CB-Ext</td>
<td>SAML V2.0 Channel Binding Extensions Version 1.0</td>
<td>sstc-saml-channel-binding-ext-v1.0.xsd</td>
</tr>
<tr>
<td>MacedirEntityCat</td>
<td>The Entity Category SAML Entity Metadata Attribute Type</td>
<td></td>
</tr>
</tbody>
</table>