A profile for distributed SAML metadata management

Version: 7 ("Gruesome Gorilla")

Editor: Leif Johansson
Contributors: Scott Cantor, Ian Young, RL "Bob" Morgan, Nate Klingenstein

0. Introduction

SAML metadata [saml-metadata-2.0-os.pdf] has proven very useful in large-scale heterogeneous SAML deployments, providing a basis for shared trust and configuration management. This document proposes a profile for management of SAML metadata to extend its utility. Benefits of deploying this profile are:

- easy discovery of published metadata based on EntityID
- simple runtime entity key management
- support for trust communities and reputation services

A metadata aggregator is an entity which collects, proofs and publishes metadata from an alternate location than one associated with any one entity. A metadata aggregate is a signed SAML metadata document containing an EntitiesDescriptor element.

A multi-party federation (or trust community) will maintain multiple metadata aggregates, for instance an aggregate of all IdPs in a federation signed by a high-assurance key (eg associated with a small-scale Ad-Hoc PKI). Such an aggregate could be used to support IdP discovery.

In addition to publishing a metadata aggregate of all the entities a federation may choose to publish aggregates of SPs based on minimal required level of assurance (of the metadata management itself), or aggregates of IdP based on the minimally asserted level of assurance of the constituent identities.

1. Metadata contents requirements

Entity identifiers [TBD] MUST be valid URLs using either the http or https schemes. The Name attribute of an EntitiesDescriptor element SHOULD be a valid URL using either the http or https schemes.

In the case of an https:-scheme URL, the metadata consumer MUST ignore any failures detected while validating the certificate chain of the TLS connection. Trust in metadata is conveyed using signatures on the metadata rather than through the transport by which the metadata was retrieved.

Any metadata role supporting TLS or digital signatures MUST contain at least one applicable KeyDescriptor containing either a certificate or public key. A relying party MUST accept TLS authentication or digital signing using any of those keys. If a certificate is present, its contents other than the public key MUST be ignored by the metadata consumer.

Any metadata role supporting encryption MUST contain at least one applicable KeyDescriptor containing one or more public keys, either in the form of RSAKeyValue elements or contained within X.509 certificates included as X509Data elements. In the case of X.509 certificates, the metadata consumer ignores all components of the certificate other than the public key.

KeyName elements MAY be included to help in identifying keys. [NOTE - isn't this a repeat of what SAMLMeta already says?]

Entities (eg an SP or IdP) SHOULD support generation of self-signed certificates (or keys) unless another set of certificates or keys was explicitly provided. Metadata producers SHOULD allow the choice of a single key or separate keys for singing and encryption to be configured.

The metadata MUST specify either a cacheTTL or expiry time and this cache policy MUST be respected by consumers of the metadata. If the metadata is signed the signing key SHOULD be signed by a metadata signing CA.

2. Metadata management requirements

In this profile there are two ways to publish metadata: per entity trusted publishing or aggregator-based publishing. Similarly there are two ways to establish trust in the metadata signatures: based on metadata signing certificates (cf below) together with a traditional PKI
or using out-of-band certificates used as a form of pre-shared keys for signature validation.

Both forms of publishing can be combined with both forms of trust establishment.

All entities mustsupport per-entity publishing. All metadata consumers must support both forms of publishing and signature trust establishment.

2.1 Metadata Publishing

All metadata MUST conform to the requirements in section 1. A metadata consumer SHOULD cache any downloaded metadata and retain the cached metadata in accordance with the cache policy expressed in the metadata cacheTTL and or expiry time.

2.1.1 Per Entity Publishing

An HTTP GET-request to the EntityID URL MUST return SAML metadata [saml-metadata-2.0-os.pdf] for the entity encoded as the <TBD> mime-type.

2.1.2 Aggregated Publishing

An HTTP GET-request to the URL in the Name attribute of the EntitiesDescriptor element MUST return SAML metadata [saml-metadata-2.0-os.pdf] for the entity encoded as the <TBD> mime-type.

2.2 Metadata Trust

A metadata consumer MUST establish trust in the metadata by validating the signature on the metadata, regardless of how it was obtained (eg as an aggregate or as metadata for a single entity). Care should be taken not to introduce unnecessary complication into the metadata signature validation process.

No requirements are placed on the SubjectDN of certificates by this specification and metadata consumers MUST NOT rely on any part of the SubjectDN of the certificate which signed the metadata.

2.2.1 PKIX-based trust using Metadata signing certificates

A metadata signing certificate is any certificate which supports XML digital signatures and which contains a SubjectAltName extension containing the names of the entities or collections for which the key is a trusted signer. A metadata signing CA is any CA which issues metadata signing certificates.

If the metadata contains a single entity then the signature MUST NOT be trusted by the metadata consumer unless the SubjectAltName extension contains a uniformResourceLocator value which exactly matches the EntityID.

If the metadata contains a collection (eg a EntitiesDescriptor element) then the signature MUST NOT be trusted by the metadata consumer unless the SubectAltName extension contains a uniformResourceLocator value which exactly matches the Name attribute of the EntitiesDescriptor element.

In both cases SubjectAltName values of any other type than uniformResourceIdentifier (URL) MUST be silently ignored.

To ensure PKIX interoperability metadata signature certificates SHOULD contain a critical KeyUsage extension with the digitalSignature bit set. Consumers of metadata signatures MUST NOT check or enforce this bit.

2.2.2 Out-of-band Pre-Shared Certificate Trust

A metadata consumer MUST allow a certificates to be explicitly trusted to sign metdata for a selecte EntityID or EntitiesDescriptor Name URL. A metadata consumer SHOULD allow multiple certificates to be simultaneously trusted for a single entity/aggregate in order to support key rollover.

If the metadata contains a signature using any of the trusted certificates then the metadata consumer MUST trust the metadata regardless of whether the certificate is a metadata signing certificate in the sense of 2.2.1 or not.

In this case the metadata consumer MUST validate the expiration time of the certificate but MUST NOT attempt to validate any certificate chain which may be associated with the certificate.