Metadata Aggregation Practice Statement

This document is still in its draft form, any comments welcome

Introduction

The main function of eduGAIN is to act as a trusted exchange service of information required for interfederation to work. This document describes the methods used to facilitate interfederation based on SAML and must be seen as an addition to the eduGAIN SAML Profile document [eduGAIN-Profile].

The current mode of operations of the eduGAIN SAML profile is to collect entities from participant federations provided in the form of federation metadata feeds, combine them into a single eduGAIN aggregate and republish. The aggregation process also serves as a validation service in order to ensure that the resulting global eduGAIN matadata aggregate conforms to all required standards.

This central component of eduGAIN SAML service is called Metadata Distribution Service (MDS).

Technical operational details about metadata signing, publication and other procedures can be found in the eduGAIN Operational Practice statemer Document [eduGAIN-OPS].

Terms

The terms defined below are a required extension of the terminology defined in [eduGAIN-Profile]. The reader should consult both dictionaries for a complete picture.

federation metadata feed	A SAML metadata file originating from a participant federation acting as a SAML Metadata Producer
eduGAIN matadata aggregate	A SAML metadata file obtained as an aggregate of federation metadata feeds according to the procedures described in this document
federation metadata channel	A location (in the form of http/https URL) pointing to the distribution source of the federation metadata feed

Source of metadata

MDS bases its aggregation function of information provided by each participant Federation as specified in [eduGAIN-Profile]:

- A federation metadata channel
- An RSA public key with which the metadata metadata feed document will be signed. This will normally be made available in the form of an X.509 certificate.
- The registrationAuthority attribute value to be associated with the federation metadata feed

This information needs to be registered with eduGAIN OT in a trust preserving way as described in [eduGAIN-OPS].

In order to eliminate unnecessary traffic, the http/https server serving the federation metadata feed location SHOULD support the Conditional GET Request, this way signalling that the federation metadata feed has not been changed.

Metadata acquisition and validation

General

After a successful verification (as described further down), each federation metadata feed is saved for possible future use.

If a saved federation metadata feed copy exists and it also follows from the Conditional GET Request that the feed has not changed, the saved copy is being used for further processing.

A federation metadata channel which cannot deliver a document (fetched or from cache) that passes all of the required tests is regarded as empty.

Verification of origin

As specified by the [eduGAIN-Profile] in order to assure metadata integrity and originality, each federation metadata feed MUST be signed as specified in [SAMLMeta]. This signature made with the key matching the one supplied to the eduGAIN OT is the only element on which trust is based. In particular MDS does not use trust that might be derived from an https endpoint details.

Metadata signature verification is done against the public key alone. If the public key for the federation metadata feed channel is supplied in the form of an X.509 certificate, other aspects of the certificate such as its expiry date do not form part of signature verification. This is in accordance with the SAML metadata interoperability profile. In particular an expired certificate will still be used for the verification purpose.

	condition evaluated	**reason**
S1	The signature exists and is valid	eduGAIN-profile
S2	The signature can be validated with the public key configured for the federation metadata channel	eduGAIN-profile
S3	The signature RSA key size is at least 2048-bit	eduGAIN-profile

In the verification process the following criteria of the XML signature are also considered. However, at the moment they are not considered to be fatal errors. (some items on this list may be moved to the table above if eduGAIN policy makes them mandatory)

- The signature was made using an explicit ID reference, not an empty reference.
- The signature reference refers to the document element (this helps to avoid "wrapping attacks").
- The digest algorithm is at least as strong as SHA-256. Specifically, MD5 and SHA-1 are not permitted as digest algorithms.
- The signature method is RSA with an associated digest at least as strong as SHA-256. Specifically, MD5 and SHA-1 are not permitted as digest algorithms.
- The signature's transforms contain only permissible values:
    - Enveloped signature
    - Exclusive canonicalisation with or without comments

Verification of metadata validity

After a positive verification of integrity and originality (as described in the previous section), the following validity verification steps are performed.

Verification of the document as a whole:

	Condition Evaluated	**Reason**
A1	the document element is md:EntitiesDescriptor	
A2	all required namespaces are declared, that is xmlns:md, xmlns:mdrpi, xmlns:ds.	
A3	if md:EntitiesDescriptor contains md:Extensions element with mdrpi:PublicationInfo element in which the publisher attribute is given	
A4	validUntil attribute in EntitiesDescriptor element exists, can be converted to a time value and it does not point to the past	SAML lines: 348; 316
A5	validUntil attribute with a value not earlier than 120 hours (5 days) and not later than 2304 hours (28 days) after the creationInstant	eduGAIN-profile

A6	the fetched document schema-validates against following SAML metadata schemas: • saml-schema-metadata-2.0.xsd - namespace urn:oasis:names:tc:SAML:2.0:metadata • saml-schema-assertion-2.0.xsd - namespace urn:oasis:names:tc:SAML:2.0:assertion • saml-metadata-rpi-v1.0-csd01.xsd - namespace urn:oasis:names:tc:SAML:metadata:rpi • shibboleth-metadata-1.0.xsd - namespace urn:mace:shibboleth:metadata:1.0 • sstc-metadata-attr.xsd - namespace sstc-saml-metadata-ui-v1.0.xsd - namespace urn:oasis:names:tc:SAML:metadata:ui • sstc-saml-idp-discovery.xsd - namespace urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol • sstc-saml-metadata-algsupport.xsd - namespace urn:oasis:names:tc:SAML:metadata:algsupport • xml.xsd - namespace http://www.w3.org/XML/1998/namespace • xmldsig-core-schema.xsd - namespace http://www.w3.org/2000/09/xmldsig# • xenc-schema.xsd - namespace http://www.w3.org/2001/04/xmlenc# • ws-addr.xsd - namespace http://www.w3.org/2005/08/addressing • ws-securitypolicy-1.2.xsd - namespace http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702 • ws-authorization.xsd - namespace http://docs.oasis-open.org/wsfed/authorization/200706 • ws-federation.xsd - namespace http://docs.oasis-open.org/wsfed/federation/200706 • MetadataExchange.xsd - namespace http://schemas.xmlsoap.org/ws/2004/09/mex • oasis-200401-wss-wssecurity-utility-1.0.xsd - namespace http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd • oasis-200401-wss-wssecurity-secext-1.0.xsd - namespace http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	

For each md:EntityDescriptor element the following verification is performed:

	Condition Evaluated	**Reason**
E1	entityID attribute value has no space characters, starts with http:// or https:// or urn: and must be unique within given feed	SAMLmeta, ^anyURI
E2	md:Extensions element with mdrpi:RegistrationInfo is defined and registrationAuthority attribute matches the value registered with the eduGAIN OT for a given federation	eduGAIN-profile
E3	if within md:ContactPerson element any of the following elements is declared: GivenName, Surname, EmailAddress, TelephoneNumber - its values must not be empty	SAMLmeta, ^string
E4	if md:Organization element is declared with md:OrganizationDisplayName and/or md:OrganizationName and/or md: OrganizationURL elements then values of these elements must not be empty	SAMLmeta, ^anyURI ^string

^anyURI - see [SAML] 1.3.2

^string - see [SAML] 1.3.1

For each role descriptor element declared under md:EntityDescriptor the following verification is performed:

	Condition Evaluated	**Reason**
R1	md:IDPSSODescriptor element must have a signing certificate (ds:KeyDescriptor/ds:KeyInfo/ds:X509Data/ds:X509Certificate)	
R2	if md:Extentions element with md:UIInfo exists: • mdui:Keywords, mdui:DisplayName, mdui:Description elements if declared must not be empty • mdui:Logo element if is declared must have a value starting with one of: http://, https:// or data:image • mdui:PrivacyStatementURL element if declared must have value starting with http:// or https://	
R3	if md:Extentions element with md:DiscoHints exist: • mdui:IPHint, mdui:DomainHint, mdui:GeolocationHint elements if declared must not empty • mdui:GeolocationHint element if declared must not be empty and must start with geo: prefix	

Resulting eduGAIN metadata aggregate

Federation metadata feeds are combined into a single collection - the eduGAIN metadata aggregate as described in detail later. If an md:EntityDescriptor /@entityID value appears in more than one federation metadata feed, the resulting collection will contain only one of the entities; the others will be discarded. MDS does not attempt to merge or otherwise combine the clashing entity descriptions. See the technical details for a description of the collision handling algorithm.

For each entity document the following are removed:

- */@xml:base
- md:EntityDescriptor/@ID
- md:EntityDescriptor/@validUntil
- md:EntityDescriptor/@cacheDuration

The eduGAIN metadata aggregate's md:EntitiesDescriptor element sets the following attributes:

- name is set to http://edugain.org
- validUntil is set 96 hours into the future
- cacheDuration is set to 6h
- ID is based on the time of its generation and has the format “eduGAIN” followed by the complete UTC date/time value (YYYYMMDDThhmmssZ)

The eduGAIN metadata aggregate is signed in conformance to the signature profile described in section 3.1 of [SAMLMeta]. In particular the signature:

- is enveloped - http://www.w3.org/2000/09/xmldsig#enveloped-signature
- contains ds:Reference containing a URI reference to the document element's ID attribute
- uses SHA-256 digest method - http://www.w3.org/2001/04/xmlenc#sha256
- uses RSA + SHA-256 signature method - http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
- uses exclusive canonicalisation - http://www.w3.org/TR/2001/REC-xml-c14n-20010315

Alerts and information

In the case when

- a federation metadata feed is unavailable (the corresponding federation feed channel is not responding)
- a federation metadata feed does not validate correctly

an alert is raised and delivered to the Operational Team. An error status is set on the eduGAIN status page https://technical.edugain.org/status and the cause of the error is displayed in the details section. The remaining cache time is also displayed. The status is also available through the eduGAIN access API, as described on: https://technical.edugain.org/monitoring. If the error condition persists reminder messages are sent in the intervals of 6 hours. If the federation metadata feed can be accessed/validated again, a recovery message is delivered to the eduGAIN OT.

During every aggregation run the validUntil timer for each of the federation metadata feeds is performed.

- If the remaining validity period is below 96 and above 12 hours an alert is raised once a day at 14 hour UTC
- If the remaining validity period is below 12 and above 6 hours an alert is raised every second hour
- If the remaining validity period is below 6 hours an alert i raised every hour

Detailed technical description

Metadata acquisition

Federation public keys, federation feed channel locations (metadata URL), registrationAuthority strings are stored in the eduGAIN database.

Aggregation process is performed in the following steps:

- all federations with the status “in production” are selected from the eduGAIN database
- for each federation its metadata URL is used to access federation metadata feed
- the metadata URL is contacted by presenting If-None-Match and If-Modified-Since header values from the last successful metadata fetching process (conditional GET support)
- the response 304 means that metadata was not modified - in this case the latest saved copy is used in aggregation process
- the response 200 means that a new metadata feed is available
    - the eduGAIN validator is run against any new metadata feed
    - any feed error generated by the eduGAIN validator triggers the appropriate report, the offending metadata is rejected and the last successful saved copy is used instead if it is still valid
    - any successfully checked metadata feed is saved locally

Metadata validation

Each freshly downloaded federation metadata feed is processed in order to verify integrity and originality and the adherence to all required standards and policy conditions.

Signature verification is handled with the Shibboleth Metadata Aggregator v. 0.9.2

Schema conformance validation is handled with the Shibboleth Metadata Aggregator v. 0.9.2

Additional conditions, in particular those defined by the [eduGAIN-Profile] are handled by eduGAIN specific code in the eduGAIN validator implemented in Python with lxm and OpenSSL modules.

Metadata combination and collision handling

All valid federation metadata feeds are passed to the aggregator in a sequence ordered according to the date when federations have started to supply data to eduGAIN. During aggregation the first occurrence of a given entityID will be used in the resulting eduGAIN metadata aggregate, any of the following occurrences will be discarded.

It should be noted that this algorithm makes it possible that an entity being served by one federation will be later replaced by its version from another federation if this latter federation comes first in the processing order.

Metadata aggregation is performed with pyFF (currently 0.10.0dev)

Acknowledgment

This document borrows heavily from Ian Young's https://gist.github.com/iay/7486653

References

[SAML] https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAMLMeta] https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[eduGAIN-Profile]

[eduGAIN-OPS]