Table of Contents

Version

0.01

Date

2018-04-13

Status

In Progress

Required Information

Document identifier: TBD

1. Introduction

This profile specifies behavior and options that deployments of the SAML V2.0 Web Browser SSO Profile [SAML2Prof], and related profiles, are required or permitted to rely on. This revision, the first major rewrite of this material, reflects the input of many experienced implementers and deployers of this technology and best practice developed over 15 years of experience with varied approaches. While it has an emphasis on highly-scaled multi-lateral federation deployments involving thousands of Identity Providers (IdPs) and Service Providers (SPs), most of these requirements are applicable to virtually any significant deployment of SAML SSO.

The requirements specified are in addition to all normative requirements of the underlying Web Browser SSO and Single Logout profiles [SAML2Prof], as modified by the Approved Errata [SAML2Err], and readers are assumed to be familiar with all relevant reference documents. Any such requirements are not repeated here except where deemed necessary to highlight a point of discussion or draw attention to an issue addressed in errata, but remain implied.

Nothing in this profile should be taken to imply that disclosing personally identifiable information, or indeed any information, is required from an IdP with respect to any particular SP. That remains at the discretion of applicable settings, user consent, or other appropriate means in accordance with regulations and policies. However, this profile does obligate IdPs to honor certain key signals from an SP in the area of subject identification and requires successful responses to include specific SAML Attributes under certain conditions. Failure is always an option.

Note that SAML features that are optional, or lack mandatory processing rules, are assumed to be optional and out of scope of this profile if not otherwise precluded or given specific processing rules.

This profile also addresses only the direct participants in the covered profiles, and does not include processing requirements related to proxying. While common, proxying introduces significant differences that are appropriately addressed only in a dedicated fashion and not as an incidental element.

2. Notation and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD

NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification uses the following typographical conventions in text: `<ns:Element>`, `Attribute`, **Datatype**, `OtherCode`. The normative requirements of this specification are individually labeled with a unique identifier in the following form: **[SDP-EXAMPLE01]**. All information within these requirements should be considered normative unless it is set in *italic* type. Italicized text is non-normative and is intended to provide additional information that may be helpful in implementing the normative requirements.

2.1. References to SAML 2.0 specification

When referring to elements from the SAML 2.0 core specification [SAML2Core], the following syntax is used:

- `<samlp:ProtocolElement>` - for elements from the SAML 2.0 Protocol namespace.
- `<saml:AssertionElement>` - for elements from the SAML 2.0 Assertion namespace.

When referring to elements from the SAML 2.0 metadata specification [SAML2Meta], the following syntax is used:

- `<md:MetadataElement>`

When referring to elements from the SAML 2.0 Metadata Extensions for Login and Discovery User Interface specification [MetaUi], the following syntax is used:

- `<mdui:MetadataElement>`

When referring to elements from the SAML 2.0 Metadata Extension for Entity Attributes specification [MetaAttr], the following syntax is used:

- `<mdattr:MetadataElement>`

When referring to elements from the SAML V2.0 Asynchronous Single Logout Protocol Extension specification [SAML2ASLO], the following syntax is used:

- `<aslo:Element>`

When referring to elements from the XML-Signature Syntax and Processing Version 1.1 WWWC Recommendation [XMLSig], the following syntax is used:

- `<ds:Element>`

2.2. Terminology

The abbreviations IdP and SP are used below to refer to Identity Providers and Service Providers in the sense of their usage within the SAML Browser SSO Profile and Single Logout profiles.

Whether explicit or implicit, all the requirements in this document are meant to apply to deployments of SAML profiles and may involve explicit support for requirements by SAML-

implementing software and/or supplemental support via application code. Deployments of a Service Provider may refer to both stand-alone implementations of SAML, libraries integrated with an application, or any combination of the two. It is difficult to define a clear boundary between a Service Provider and the application/service it represents, and unnecessary to do so for the purposes of this document.

3. Common Requirements

This section includes material of general significance to both IdPs and SPs. Subsequent sections provide guidance specific to those roles.

3.1. General

3.1.1. Clock Skew

[SDP-G01]

Deployments MUST allow for a minimum of three (3) and a maximum of five (5) minutes of clock skew — in either direction — when interpreting `xsd:dateTime` values in assertions and when enforcing security policies based thereupon.

The following is a non-exhaustive list of items to which this directive applies: `NotBefore`, `NotOnOrAfter`, *and* `validUntil` *XML attributes found on* `<saml:Conditions>`, `<saml:SubjectConfirmationData>`, `<samlp:LogoutRequest>`, `<md:EntityDescriptor>`, `<md:EntitiesDescriptor>`, `<md:RoleDescriptor>`, *and* `<md:AffiliationDescriptor>` *elements.*

3.1.2. Data Size

[SDP-G02]

Unless otherwise specified, deployments MUST limit the size of all element and attribute content they produce to 256 characters. This applies in particular to the values within `<saml:NameID>` and `<saml:AttributeValue>` elements.

3.1.3. Document Type Definitions

[SDP-G03]

Deployments MUST NOT produce any SAML protocol message that contains a (DTD) Document Type Definition. Deployments SHOULD reject messages that contain them.

3.1.4. SAML entityIDs

[SDP-G04]

Deployments MUST be named via an absolute URI whose total length MUST NOT exceed 256 characters.

An entityID SHOULD be chosen in a manner that minimizes the likelihood of it changing for political or technical reasons, including for example a change to a different software implementation or hosting provider.

3.2. Metadata and Trust Management

3.2.1. Metadata Consumption and Use

[SDP-MD01]

Deployments MUST provision their behavior in the following areas based solely on the consumption of SAML Metadata [SAML2Meta] on an automated, periodic or real-time basis using (where applicable) the processing rules defined by the SAML Metadata Interoperability profile [SAML2MDIOP]:

- indications of support for Browser SSO and Single Logout profiles
- selection, determination, and verification of SAML endpoints and bindings
- determination of the trustworthiness of XML signing keys and TLS client and server certificates
- selection of XML Encryption keys
- determination of subject identifier SAML Attribute(s) to provide (per `[SAML2SubjId]`)
- optional signing of assertions via the `WantAssertionsSigned` flag
- optional enforcement of request signing via the `AuthnRequestsSigned` flag

Deployments MUST NOT require out of band communication or coordination for the management of any behavior by peers included within the enumerated areas identified above. Deployments MAY of course rely on additional sources of policy, including other metadata content, in order to make determinations whether to successfully interact with peers or refuse to do so.

[SDP-MD02]

Consumption of metadata MUST be contingent on verification of a signature (STRONGLY RECOMMENDED) or TLS server certificate. The key ultimately used to establish trust in metadata MUST NOT itself appear within the same metadata in a `<md:KeyDescriptor>` element.

In most cases, the previous requirement implies that a key communicated via metadata may not also be used to sign and verify the same metadata, but it is possible to envision scenarios in which this may happen if metadata verification relies on a chain of certificates signed by an ultimately trusted Certificate Authority. However, it must be possible to seamlessly communicate new keys without necessarily changing the key used to establish trust in the metadata, which implies some level of indirection is required.

Metadata Validity

[SDP-MD03]

Metadata without a `validUntil` attribute on its root element MUST be rejected. Metadata whose root element's `validUntil` attribute extends beyond a deployer- or community-imposed threshold MUST be rejected.

These are critical (but very simple to implement) requirements for secure application of `[SAML2MDIOP]` because it is the method by which keys are revoked and the window of revocation is established.

3.2.2. Metadata Production

[SDP-MD04]

Deployments MUST have the ability to provide SAML metadata capturing their requirements and characteristics in the areas identified above in a secure fashion, the specifics of which will necessarily vary by context and community. The use of services offering third-party validation, curation, signing, and publishing of metadata is a recommended practice.

This profile does not mandate any specific automated support for the production of metadata by a deployment. In fact, automatic generation of metadata has a strong tendency to undermine the correct functioning of peer deployments in the face of key rollover or changes to endpoints or other software features because it tends to change too suddenly to accommodate a graceful transition between states.

[SDP-MD05]

Metadata MAY include content indicating support for profiles or features beyond the bounds of this profile, but metadata MUST NOT contain content that advertises profile support or features that aren't supported or that have not been deliberately and intentionally configured by a deployment.

As an example, deployments that lack support for, or have not tested and integrated an implementation's support for the HTTP-Artifact binding [SAML2Bind] must omit such endpoints.

Keys and Certificates

[SDP-MD06]

Public keys used for signing, encryption, and TLS client and server authentication MUST be expressed via X.509 certificates included in metadata via `<md:KeyDescriptor>` elements.

These certificates SHOULD be long-lived and self-signed. To avoid problems with non-conforming SAML implementations, certificates in metadata SHOULD NOT be expired (though deployments complying with this profile MUST accept expired certificates by virtue of [SAML2MDIOP]).

[SDP-MD07]

RSA public keys MUST be at least 2048 bits in length. At least 3072 bits is RECOMMENDED for new deployments.

[SDP-MD08]

EC public keys MUST be at least 256 bits in length.

[SDP-MD09]

Certificates used MUST NOT be signed with an MD5-based signature algorithm and SHOULD NOT be signed with a SHA1-based signature algorithm.

[SDP-MD10]

By virtue of the profile's overall requirements, an IdP's metadata MUST include at least one signing certificate (that is, an `<md:KeyDescriptor>` with no `use` attribute or one set to `signing`), and an SP's metadata MUST include at least one encryption certificate (that is, an `<md:KeyDescriptor>` with no `use` attribute or one set to `encryption`).

Discovery and User Interface Elements

[SDP-MD11]

Metadata MUST include an `<mdui:UIInfo>` element as defined in [MetaUI] containing at least the child elements `<mdui:DisplayName>`, `<mdui:Logo>`, `<mdui:InformationURL>`, and `<mdui:PrivacyStatementURL>`.

[SDP-MD12]

The content of the `<mdui:Logo>` element MUST be either an `https` URL or an in-line image embedded in a `data` URI element.

[SDP-MD13]

At least one `<mdui:Logo>` element MUST have a `height` attribute of `60` and a `width` attribute of `80`.

An entity SHOULD include an `<mdui:Logo>` element with a `height` attribute of `16` and a `width` attribute of `16`.

Any logo referenced by an `<mdui:Logo>` element MUST be in PNG format with a transparent background.

3.3. Cryptographic Algorithms

[SDP-ALG01]

Deployments MUST support, and use, the following algorithms when communicating with peers in the context of this profile. Where multiple choices exist, any of the listed options may be used. The profile will be updated as necessary to reflect changes in government and industry recommendations regarding algorithm usage.

- Digest
  - `http://www.w3.org/2001/04/xmlenc#sha256` [XMLEnc]
- Signature
  - `http://www.w3.org/2001/04/xmldsig-more#rsa-sha256` [RFC4051]
  - `http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256` [RFC4051]
- Block Encryption
  - `http://www.w3.org/2009/xmlenc11#aes128-gcm` [XMLEnc]
  - `http://www.w3.org/2009/xmlenc11#aes192-gcm` [XMLEnc]
  - `http://www.w3.org/2009/xmlenc11#aes256-gcm` [XMLEnc]
- Key Transport
  - `http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p` [XMLEnc]
  - `http://www.w3.org/2009/xmlenc11#rsa-oaep` [XMLEnc]

The following default digest algorithm MUST be used in conjunction with the above key transport algorithms (the default mask generation function, MGF1 with SHA1, MUST be used):

- http://www.w3.org/2001/04/xmlenc#sha256 [XMLEnc]

This profile cannot preclude the use of other algorithms when communicating with peers outside the scope of this profile, but the other algorithms in common use are generally considered to be weakening (e.g., SHA-1) or broken outright (e.g., RSA PKCS#1.5). Note that the use of AES-CBC block encryption algorithms remains widespread at the time of authoring, but are known to be broken [XMLEncBreak].

4. SP Requirements

4.1. Web Browser SSO

[SDP-SP01]

SPs MUST support the Browser SSO Profile [SAML2Prof], as updated by the Approved Errata [SAML2Err], with behavior, capabilities, and options consistent with the additional constraints specified in this section.

4.1.1. Requests

Binding

[SDP-SP02]

The HTTP-Redirect binding [SAML2Bind] MUST be used for the transmission of `<samlp:AuthnRequest>` messages.

[SDP-SP03]

Requests MUST NOT be issued inside an HTML frame or via any mechanism that would require the use of third-party cookies by the IdP to establish or recover a session with the User Agent. This will typically imply that requests must involve a full-frame redirect, in order that the top level window origin be associated with the IdP.

Request Content

[SDP-SP04]

The `<samlp:AuthnRequest>` message MUST either omit the `<samlp:NameIDPolicy>` element (RECOMMENDED), or the element MUST contain an `AllowCreate` attribute of "true" and MUST NOT contain a `Format` attribute.

[SDP-SP05]

The `<samlp:AuthnRequest>` message MUST NOT contain a `<saml:Subject>` element.

This is a relatively unused feature that is supported by few IdPs.

[SDP-SP06]

The message SHOULD contain an `AssertionConsumerServiceURL` attribute and MUST NOT contain an 'AssertionConsumerServiceIndex' attribute (i.e., the desired endpoint MUST be the default, or identified via the `AssertionConsumerServiceURL` attribute).

[SDP-SP07]

The `AssertionConsumerServiceURL` value, if present, MUST match an endpoint location expressed in the SP's metadata exactly, without requiring URL canonicalization/normalization.

As an example, the SP MUST NOT use a hostname with port number (such as https://sp.example.com:443/acs) in its request and without (such as https://sp.example.com/acs) in its metadata.

Authentication Contexts

[SDP-SP08]

An SP that does not require a specific `<saml:AuthnContextClassRef>` value MUST NOT include a `<samlp:RequestedAuthnContext>` element in its requests.

An SP that requires specific `<saml:AuthnContextClassRef>` values MUST specify the allowable values in a `<samlp:RequestedAuthnContext>` element in its requests, with the `Comparison` attribute set to `exact`.

An SP SHOULD NOT request a `<saml:AuthnContextClassRef>` value in the absence of a shared understanding between itself and the IdP regarding its definition.

4.1.2. Responses

Binding

[SDP-SP09]

SPs MUST support the HTTP-POST binding for the receipt of `<samlp:Response>` messages. Support for other bindings is OPTIONAL.

[SDP-SP10]

The endpoint(s) at which an SP supports receipt of `<samlp:Response>` messages MUST be protected by TLS/SSL.

XML Encryption

[SDP-SP11]

SPs MUST support decryption of `<saml:EncryptedAssertion>` elements. Support for other encrypted constructs is OPTIONAL.

Error Handling

[SDP-SP12]

SPs MUST gracefully handle error responses containing `<samlp:StatusCode>` other than `urn:oasis:names:tc:SAML:2.0:status:Success`.

[SDP-SP13]

The response to such errors MUST direct users to appropriate support resources offered by the SP or alternatively to the `errorURL` attribute in an IdP's metadata.

Forced Re-Authentication

[SDP-SP14]

SPs that include a `ForceAuthn` attribute of `true` in their requests SHOULD test the currency of the `AuthnInstant` element in the received assertions to verify the currency of the authentication event.

This is necessary because clients can freely generate requests that do not specify this attribute, potentially bypassing the SP's intent.

4.1.3. Subject Identification

NameID Formats

[SDP-SP15]

SPs MUST NOT require the presence of a `<saml:NameID>` element and MUST NOT rely on the content of this element for long term identification of subjects; `<saml:Attribute>` elements MUST be used for this purpose in the manner detailed below.

Subject Identifiers

[SDP-SP16]

If an SP requires persistent tracking/identification of its users (as most do), then it MUST support one or both of the SAML Attributes defined by [SAML2SubjId] for this purpose.

If an SP requires coordination and/or correlation of user activity between itself and other SPs, then the SAML Attribute named `urn:oasis:names:tc:SAML:attribute:subject-id` *is appropriate. Otherwise the SAML Attribute named* `urn:oasis:names:tc:SAML:attribute:pairwise-id` *can be used.*

[SDP-SP17]

SPs MAY support legacy or historical `<saml:NameID>` and `<saml:Attribute>` content for compatibility reasons but MUST NOT require their use.

Subject Identifier Requirements Signaling

[SDP-SP18]

An SP MUST represent its identifier requirements in its SAML metadata, consistent with the Requirements Signaling mechanism defined in [SAML2SubjId].

Identifier Scoping

[SDP-SP19]

SPs MUST prevent unintended identifier collisions in the values asserted by different IdPs, and the required identifier types, per [SAML2SubjId], are "scoped" via a DNS-like syntax to help fulfill this requirement.

[SDP-SP20]

SPs MUST associate identifier scopes with IdPs such that only authorized IdPs may assert identifiers with particular scopes for particular purposes.

For example, if the `example.com` scope is bound to the IdP named `http://idp.example.com/saml`, it should be generally disallowed for any other IdP to assert an identifier in that scope. Note that this is not a 1:1 relationship; it may frequently happen that multiple IdPs may assert a given scope, or an IdP may assert identifiers in multiple scopes, but the rules for this should be explicit and enforced.

Displayable Identifiers

The required identifier types above are opaque, unknown to users in most cases, and unsuitable for display.

[SDP-SP21]

SPs requiring the display of identifiers to users, the identification of other users via searching, selection, etc., and similar use cases SHOULD rely on additional suitable SAML Attributes such as ([X500SAMLattr]):

- `urn:oid:0.9.2342.19200300.100.1.3` (mail)
- `urn:oid:2.16.840.1.113730.3.1.241` (displayName)
- `urn:oid:2.5.4.42` (givenName)
- `urn:oid:2.5.4.4` (sn)

Note that most standardized Attributes of this sort tend to be defined as multi-valued.

4.1.4. Attribute Value Constraints

[SDP-SP22]

When consuming SAML Attributes with standardized definitions in external specifications, SPs MUST NOT impose constraints beyond the definitions of those attributes.

For example, the definition of the `mail` attribute (in SAML, `urn:oid:0.9.2342.19200300.100.1.3`) explicitly allows for multiple values, so an SP that consumes it for some purpose must necessarily allow for that possibility.

4.1.5. Usability

Silo-oriented, multi-tenant approaches to federated application deployment create an inherent friction with the intended design of the web, user behavior and experience, and the needs of collaboration inherent in many applications. SSO, when integrated poorly, can negatively impact usability, and the following sections, while not strictly matters of SAML interoperability, have a significant effect on the perception of the system as a whole and on the successful adoption of SSO, regardless of the protocol.

The web inherently operates on the basis of *addressability* of resources; that is, users expect to be able to access a piece of information or an application function directly, without regard for their identity, current level of access, or what is convenient for an application developer to support. This leads naturally to the ability to create bookmarks to what matters to them, and users will consistently route around attempts to force them through proxies, portals, and other artificial access paths.

At a high level, these issues fall under the term `deep linking`.

For a wide range of applications in the collaborative space, this notion is not merely convenient, but utterly essential, because such applications presume the sharing of resources with peers between organizations.

For the purposes of the following requirements, we will refer to applications that rely on the exposure of resource URLs that may be shared between users from multiple organizations as "collaborative" applications, even if their purpose may not specifically align with that term.

Support for Multiple IdPs

[SDP-SP23]

> SPs MUST allow for the possibility that any given request requiring authentication may be potentially satisfied by more than one IdP. That is, any scenario in which a piece of content, policy, configuration, or decision on the part of an application is bound to an IdP MUST be constructed in a fashion such that more than one IdP may be so bound.

This requirement flows from both the inherent requirements of collaborative applications described above, and from the simple reality that enterprises vary in their structure. Some organizations rely on more than one IdP due to administrative boundaries, but frequently contract for or access services as a single body. Thus, any presumed mapping between a contract or set of access policies and a single SAML IdP is too constraining. This constraint imposes a need for complex proxying of SSO by many organizations and must be avoided.

Deep Linking

[SDP-SP24]

> Applications SHOULD, and collaborative applications MUST, support deep linking. Deep linking implies maintaining support for such links across the boundary of a Web Browser SSO profile interaction involving any IdP necessary to complete the login process. That is, it SHOULD be possible to request a resource and (authorization permitting) have it supplied as the result of a successful Web Browser SSO profile exchange.

[SDP-SP25]

> It is RECOMMENDED that SPs support the preservation of POST bodies across a successful SSO profile exchange, subject to size limitations dictated by policy or implementation constraints.

Deep linking implies support for SP-initiated SSO, i.e., the direct generation of authentication request messages in response to unauthenticated or insufficiently-authenticated access attempts to an application as a whole, or to specific protected content. Deep linking may co-exist with support for unsolicited responses (so-called IdP-initiated SSO), but precludes its requirement.

Discovery

Deep linking also implies support for some form of IdP "discovery", the process by which an SP establishes which IdP to use on behalf of a subject. Use of IdP-initiated SSO is a common workaround for supporting discovery, but cannot be required if deep linking is supported, in addition to having other drawbacks.

A common means of discovery is the mapping of resource/application URL (typically virtual host, sometimes path) to a specific IdP. This is strongly discouraged, and is disallowed for collaborative applications, since it makes the sharing of URLs between users from multiple organizations impossible (or at best highly inconvenient).

[SDP-SP26]

SPs SHOULD consider support for the Identity Provider Discovery Service Protocol and Profile defined in [IdPDisco] as it provides a general, composable building block. SPs MAY support other mechanisms and caching solutions (e.g., cookies) as desired, to reduce the frequency of discovery.

4.2. Single Logout

[SDP-SP27]

SPs MAY support the Single Logout Profile [SAML2Prof], as updated by the Approved Errata [SAML2Err]. The following requirements apply in the case of such support.

4.2.1. Requests

Binding

[SDP-SP28]

The HTTP-Redirect binding [SAML2Bind] MUST be used for the transmission of `<samlp:LogoutRequest>` messages.

[SDP-SP29]

SPs MUST support the HTTP-Redirect [SAML2Bind] binding for the receipt of `<samlp:LogoutRequest>` messages, in the event that inbound `<samlp:LogoutRequest>` messages are supported.

[SDP-SP30]

Requests MUST NOT be issued inside an HTML frame or via any mechanism that would require the use of third-party cookies by the IdP to establish or recover a session with the User Agent. This will typically imply that requests must involve a full-frame redirect, in order that the top level window origin be associated with the IdP.

The full-frame requirement is also necessary to ensure that full control of the user interface is released to the IdP.

Request Content

[SDP-SP31]

Requests MUST be signed.

[SDP-SP32]

The `<saml:NameID>` element included in `<samlp:LogoutRequest>` messages MUST exactly match the corresponding element received from the IdP, including its element content and all XML attributes included therein.

[SDP-SP33]

The `<saml:NameID>` element in `<samlp:LogoutRequest>` messages MUST NOT be encrypted.

The normative requirement for the use of transient identifiers is intended to obviate the need for XML Encryption.

4.2.2. Responses

Binding

[SDP-SP34]

The HTTP-Redirect binding [SAML2Bind] MUST be used for the transmission of `<samlp:LogoutResponse>` messages.

[SDP-SP35]

SPs MUST support the HTTP-Redirect [SAML2Bind] binding for the receipt of `<samlp:LogoutResponse>` messages, in the event that they do not include the `<aslo:Asynchronous>` extension [SAML2ASLO] in all of their requests.

Response Content

[SDP-SP36]

Responses MUST be signed.

4.2.3. Behavioral Requirements

[SDP-SP37]

SPs MUST terminate a subject's local session before issuing a `<samlp:LogoutRequest>` message to the IdP.

This ensures the safest possible result for subjects in the event that logout fails for some reason, as it often will.

[SDP-SP38]

SPs MUST NOT issue a `<samlp:LogoutRequest>` message as the result of an idle activity timeout.

Timeout of a single application/service must not trigger logout of an SSO session because this imposes a single service's requirements on an entire IdP deployment. Applications with sensitive requirements should consider other mechanisms, such as the `ForceAuthn` attribute, to achieve their goals.

4.2.4. Logout and Virtual Hosting

[SDP-SP39]

An SP that maintains distinct sessions across multiple virtual hosts SHOULD identify itself by means of a distinct entityID (with associated metadata) for each virtual host.

A single entity can have only one well-defined `<SingleLogoutService>` endpoint per binding. Cookies are typically host-based and logout cannot typically be implemented easily across virtual hosts. Unlike during SSO, a `<samlp:LogoutRequest>` message cannot specify a particular response endpoint, so this

scenario is generally not viable.

4.3. Metadata and Trust Management

4.3.1. Support for Multiple Keys

The ability to perform seamless key migration depends upon proper support for consuming and/or leveraging multiple keys at the same time.

[SDP-SP40]

SP deployments MUST support multiple signing certificates in IdP metadata and MUST support validation of XML signatures using a key from any of them.

[SDP-SP41]

SP deployments MUST be able to support multiple decryption keys and MUST be able to decrypt `<saml:EncryptedAssertion>` elements encrypted with any configured key.

4.3.2. Metadata Content

[SDP-SP42]

By virtue of this profile's requirements, an SP's metadata MUST contain:

- an `<md:SPSSODescriptor>` role element
  - at least one `<md:AssertionConsumerService>` endpoint element
  - at least one `<md:KeyDescriptor>` element whose `use` attribute is omitted or set to `signing`
  - at least one `<md:KeyDescriptor>` element whose `use` attribute is omitted or set to `encryption`
- an `<md:Extensions>` element
  - an `<mdui:UIInfo>` extension element with previously prescribed content
  - an `<mdattr:EntityAttributes>` extension element for signaling Subject Identifier requirements with previously prescribed content

In addition, an SP's metadata MUST contain:

- an `<md:ContactPerson>` element with a `contactType` of `technical` and an `<md:EmailAddress>` element

An `<md:SingleLogoutService>` element MAY be omitted in the event that an SP either does not support the Single Logout Profile, or solely issues `<samlp:LogoutRequest>` messages containing the `<aslo:Asynchronous>` extension [SAML2ASLO].

5. IdP Requirements

5.1. Web Browser SSO

[SDP-IDP01]

IdPs MUST support the Browser SSO Profile [SAML2Prof], as updated by the Approved Errata [SAML2Err], with behavior, capabilities, and options consistent with the additional constraints specified in this section.

5.1.1. Requests

Binding

[SDP-IDP02]

IdPs MUST support the HTTP-Redirect binding [SAML2Bind] for the receipt of `<samlp:AuthnRequest>` messages.

[SDP-IDP03]

The endpoint(s) at which an IdP supports receipt of `<samlp:AuthnRequest>` messages MUST be protected by TLS/SSL.

Endpoint Verification

[SDP-IDP04]

When verifying the `AssertionConsumerServiceURL`, it is RECOMMENDED that the IdP perform a case-sensitive string comparison between the requested value and the values found in the SP's metadata. It is OPTIONAL to apply any form of URL canonicalization.

Signing

[SDP-IDP05]

If a request is signed, IdPs MUST verify the signature or fail the request. An IdP MAY handle a signature verification failure locally rather than via an error response to the SP.

[SDP-IDP06]

IdPs MUST reject unsigned requests in the event that an SP's metadata includes an `AuthnRequestsSigned` attribute set to `true` or `1`.

Forced Re-Authentication

[SDP-IDP07]

IdPs MUST ensure that any response to a `<samlp:AuthnRequest>` that contains the attribute `ForceAuthn` set to `true` or `1` results in an authentication challenge that requires proof that the subject is present. If this condition is met, the IdP MUST also reflect this by setting the value of the `AuthnInstant` value in the assertion it returns to a fresh value.

If an IdP cannot prove subject presence, then it MUST fail the request and SHOULD respond to the SP with a SAML error status.

5.1.2. Responses

Binding

[SDP-IDP08]

IdPs MUST support the HTTP-POST binding [SAML2Bind] for the transmission of `<samlp:Response>` messages.

Response Content

[SDP-IDP09]

Successful responses MUST be directly signed using a `<ds:Signature>` element within the `<samlp:Response>` element. Error responses MAY be signed.

[SDP-IDP10]

Successful responses MUST contain one and only one SAML assertion, and the assertion MUST contain exactly one `<saml:AuthnStatement>` element and MAY contain zero or one `<saml:AttributeStatement>` elements. The assertion within the response MAY be directly signed.

[SDP-IDP11]

In the event the HTTP-POST binding [SAML2Bind] is used, assertions MUST be encrypted and transmitted via a `<saml:EncryptedAssertion>` element. Information intended for the consumption of the SP MUST NOT be further encrypted via `<saml:EncryptedID>` or `<saml:EncryptedAttribute>` constructs.

5.1.3. Subject Identifiers

[SDP-IDP12]

Assertions MUST contain a `<saml:NameID>` element with the `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` Format, as defined in [SAML2Core], for the purposes of logout.

[SDP-IDP13]

IdPs MUST support one or both of the SAML Attributes defined by [SAML2SubjId] for non-transient identification of subjects. Support for both is RECOMMENDED.

Subject Identifier Requirements Signaling

[SDP-IDP14]

IdPs MUST support the metadata-based identifier requirement signaling mechanism defined in [SAML2SubjId].

[SDP-IDP15]

If an IdP cannot or will not satisfy the requirements of an SP in this respect, then it MUST fail the authentication request and SHOULD respond to the SP with a SAML error status.

[SDP-IDP16]

In the absence of any signaling by an SP, an IdP MAY supply either, both, or neither SAML Attribute, or return an error as it sees fit.

5.1.4. Attributes

[SDP-IDP17]

`<saml:Attribute>` elements MUST contain a NameFormat of `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.

This requirement ensures unique, non-conflicting naming of Attributes even in cases involving custom requirements for which no standard Attributes may exist.

[SDP-IDP18]

It is RECOMMENDED that the content of `<saml:AttributeValue>` elements be limited to a single child text node (i.e., a simple string value).

5.2. Single Logout

[SDP-IDP19]

IdPs MUST support the Single Logout Profile [SAML2Prof], as updated by the Approved Errata [SAML2Err], with behavior, capabilities, and options consistent with the additional constraints specified in this section.

The term "IdP session" is used to refer to the ongoing state between the IdP and its clients allowing for SSO. Support for logout implies supporting termination of a subject's IdP session in response to receiving a `<samlp:LogoutRequest>` or upon some administrative signal.

[SDP-IDP20]

IdPs MAY allow a subject the option to maintain their IdP session rather than unilaterally terminating it.

[SDP-IDP21]

IdPs MAY support the propagation of logout signaling to SPs.

5.2.1. Requests

Binding

[SDP-IDP22]

The HTTP-Redirect binding [SAML2Bind] MUST be used for the transmission of `<samlp:LogoutRequest>` messages, in the event that propagation is supported.

[SDP-IDP23]

IdPs MUST support the HTTP-Redirect [SAML2Bind] binding for the receipt of `<samlp:LogoutRequest>` messages.

5.2.2. Request Content

[SDP-IDP24]

Requests MUST be signed.

[SDP-IDP25]

The `<saml:NameID>` element in `<samlp:LogoutRequest>` messages MUST NOT be encrypted.

The normative requirement for the use of transient identifiers is intended to obviate the need for XML Encryption.

5.2.3. Responses

Binding

[SDP-IDP26]

The HTTP-Redirect binding [SAML2Bind] MUST be used for the transmission of `<samlp:LogoutResponse>` messages.

[SDP-IDP27]

IdPs MUST support the HTTP-Redirect [SAML2Bind] binding for the receipt of `<samlp:LogoutResponse>` messages, in the event that `<samlp:LogoutRequest>` propagation is supported.

Response Content

[SDP-IDP28]

Responses MUST be signed.

[SDP-IDP29]

The `<samlp:StatusCode>` in the response issued by the IdP MUST reflect whether the IdP session was successfully terminated.

5.3. Metadata and Trust Management

5.3.1. Support for Multiple Keys

The ability to perform seamless key migration depends upon proper support for consuming and/or leveraging multiple keys at the same time.

[SDP-IDP30]

IdP deployments MUST support multiple signing certificates in SP metadata and MUST support validation of signatures using a key from any of them.

5.3.2. Metadata Content

[SDP-IDP31]

By virtue of this profile's requirements, an IdP's metadata MUST contain:

- an `<md:IDPSSODescriptor>` role element containing an `errorURL` attribute and an appropriate URL value
    - at least one `<md:SingleSignOnService>` endpoint element

    - at least one `<md:SingleLogoutService>` endpoint element
    - at least one `<md:KeyDescriptor>` element whose `use` attribute is omitted or set to `signing`
  - an `<md:Extensions>` element
    - an `<mdui:UIInfo>` extension element with previously prescribed content

In addition, an IdP's metadata MUST contain:

- an `<md:ContactPerson>` element with a `contactType` of `technical` and an `<md:EmailAddress>` element

6. Normative References

- [RFC2119] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, March 1997. http://www.ietf.org/rfc/rfc2119.txt
- [RFC8174] IETF RFC 8174, Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words, May 2017. http://www.ietf.org/rfc/rfc8174.txt
- [RFC4051] IETF RFC 4051, Additional XML Security Uniform Resource Identifiers, April 2005. https://www.ietf.org/rfc/rfc4051.txt
- [SAML2Core] OASIS Standard, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
- [SAML2Bind] OASIS Standard, Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf
- [SAML2Prof] OASIS Standard, Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
- [SAML2Meta] OASIS Standard, Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf
- [X500SAMLattr] OASIS Committee Specification, SAML V2.0 X.500/LDAP Attribute Profile, March 2008. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-x500-cs-01.pdf
- [SAML2MDIOP] OASIS Committee Specification, SAML V2.0 Metadata Interoperability Profile Version 1.0, August 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf
- [IdPDisco] OASIS Committee Specification, Identity Provider Discovery Service Protocol and Profile, March 2008. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf
- [SAML2Err] OASIS Approved Errata, SAML Version 2.0 Errata 05, May 2012. http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf
- [XMLEnc] D. Eastlake et al. XML Encryption Syntax and Processing. W3C Recommendation, April 2013. https://www.w3.org/TR/xmlenc-core1/
- [XMLSig] D. Eastlake et al. XML-Signature Syntax and Processing, Version 1.1. W3C Recommendation, April 2013. https://www.w3.org/TR/xmldsig-core1/
- [SAML2SubjId] OASIS Working Draft, SAMLV2.0 Subject Identifier Attributes Profile Version 1.0, February 2018. https://www.oasis-open.org/committees/download.php/62438/saml-subject-id-

attr-v1.0-wd04.pdf

- [SAML2ASLO] OASIS Committee Specification, SAML V2.0 Asynchronous Single Logout Profile Extension Version 1.0, November 2012. http://docs.oasis-open.org/security/saml/Post2.0/saml-async-slo/v1.0/cs01/saml-async-slo-v1.0-cs01.pdf
- [MetaUi] OASIS Committee Specification, SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0, April 2012. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-ui/v1.0/cs01/sstc-saml-metadata-ui-v1.0-cs01.pdf
- [MetaAttr] OASIS Committee Specification, SAML V2.0 Metadata Extension for Entity Attributes Version 1.0, August 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr-cs-01.pdf

7. Non-Normative References

- [XMLEncBreak] Jager and Somorovsky, How to Break XML Encryption, October 2011. http://www.nds.rub.de/media/nds/veroeffentlichungen/2011/10/22/HowToBreakXMLenc.pdf

8. Authors' addresses

- Andreas Åkre Solberg, UNINETT, andreas.solberg@uninett.no
- Scott Cantor, Ohio State University, cantor.2@osu.edu
- Eve Maler, Sun Microsystems, eve.maler@sun.com
- Leif Johansson, Stockholm University, leifj@sunet.se
- Jeff Hodges, Neustar, Jeff.Hodges@neustar.biz
- Ian Young, ian@iay.org.uk
- Nate Klingenstein, ndk@internet2.edu
- Bob Morgan, rlmorgan@washington.edu
- Keith Wessel, kwessel@illinois.edu
- Eric Goodman, eric.goodman@ucop.edu
- Andrew Morgan, morgan@oregonstate.edu
- Judith Bush, bushj@oclc.org
- Thomas Lenggenhager, lenggenhager@switch.ch
- Alex Stuart, alex.stuart@jisc.ac.uk
- Keith Hazelton, keith.hazelton@wisc.edu
- Chris Phillips, chris.phillips@canarie.ca
- Nick Roy, nroy@internet2.edu