Provisioning and deprovisioning in an identity federation

Problem description and solution proposals

19.12.2008/Mikael.linden@csc.fi

Contents

1. Description of the context

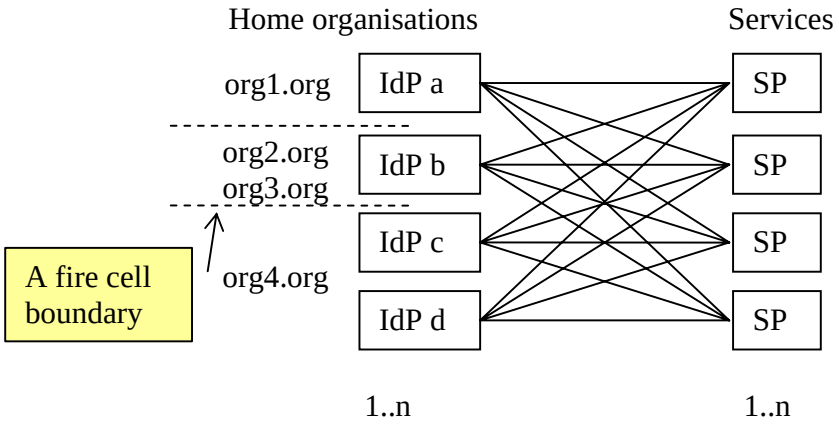

Figure 1. The technical architecture is a distributed federation (Circle of Trust) where each Identity Provider represents the identities of the end users in one or more organisations, and each organisation may have one or more Identity Providers.

In this document, it is assumed that

- the technical architecure of the federation is a distributed mesh with no centralised entities (IdP Proxy) but each Identity Provider (IdP) communicates directly with a Service Provider (SP).
- SAML 2.0 is used as the federated identity management protocol. For maximal interoperability, only features whose interoperability is tested in the Liberty Interoperable program should be used, to the extent possible.
- there may be a centralised entity (called the federation operator) who maintains the SAML 2.0 metadata of the federation and provides it to the IdPs and SPs in the federation
- each IdP may serve one or more organisations (for instance, there may be a centralised IdP “orphanage.federation.org” that acts as a collective IdP for several small organisations)
- each organization may have several IdPs (for instance, one IdP serving some basic assurance level and a hardened IdP providing strong identities and authentication). How IdP Discovery is done in this multi-IdP environment is our of scope for this document

For security reasons, “fire cells” must be implemented to ensure that one broken or compromised IdP does not affect the other IdPs and their end users. In practice, considering Figure 1, the fire cells ensure that IdP a cannot assert a valid identity for an end user in org2.org. If it tries, an SP must have a mechanism in place to detect and reject such assertions.

2. Problem description

Web single sign-on is the fundamental use scenario for SAML2.0; the user is authenticated and her/his attributes are provided to the SP when s/he logs in.

However, there are use scenarios where an end user's identity needs to be provisioned to an SP at the time when s/he becomes authorized to use the service (which may be much earlier than when s/he logs in for the first time) and deprovisioned when the authorization ceases.

Examples of provisioning

- a student's identity (e.g. membership in a course) should be provisioned from the student information system to a learning management system when s/he enrolls to the course, so that the teacher of the course can print an up-to-date list of the enrolled students at any time
- an organization has outsourced the travel expense management system and acquires it as SaaS/ASP. SAML2.0 is used for authenticating end users claiming their travel expenses, but provisioning and deprovisioning of end users to the service needs to be solved separately

2.1. Mass provisioning

Mass provisioning is a one-time task necessary when a new service is introduced in an organization and an existing user population needs to be imported to the system. Due to its nature as a one-time-task, it may consist of manual steps (e.g. running a script that populates the service with the contents of a CSV file).

For instance,

- organization has acquired a new travel expense management system as SaaS, and the users are migrated to the new service at once
- a university replaces a learning management system by a new one

2.2. Provisioning one-by-one

As opposed to mass provisioning, end users need to be provisioned to an SP one-by-one. This is necessary, for instance, when

- a new end user (e.g. an employee or a student) enters the organization and his/her identity needs to be provisioned to the services s/he is authorized to use
- an existing end user gets a new permission to use an existing service

One-by-one provisioning is expected to be a frequent procedure (e.g. every night), and manual intervention by IdP or SP administration should be avoided.

2.3. Deprovisioning one-by-one

Deprovisioning means removing or closing an end user's account in a service when s/he departs or his/her authorisation to use the service ceases otherwise.

From information security perspective, closing and end user's account in the IdP is enough to stop him/her from using WebSSO to the service. However, if the service has workflows in which the user is involved, it may be necessary to deactivate his/her account in the service, as well. For instance, if the end user has the privilege of approving travel expenses in the travel expense management system, closing his/her accounts in the IdP does not stop the other users from submitting their travel expense reports to him/her in the service.

Furthermore, the data protection directive requires that a data subject's personal data is kept up-to-date in the service, as well.

2.4. Mass deprovisioning

Mass deprovisioning is done when a service is to be shut down and all the users are removed at once. It is considered trivial and not further discussed in this document.

3. Solution proposals

The solution proposals, which are discussed in detail in the subsequent sections, are summarized in Table 1. The table shows which problems are solved by the proposed solutions.

Table 1. Summary of solution proposals. X=suits, XX=suits well

	Mass provisioning	One-by-one provisioning	One-by-one deprovisioning
On-the-fly using SAML during the first sign-on	X	XX	
Manual push	XX		
Automated push, bilateral	XX	XX	XX
Automated push, federation-assisted	XX	XX	XX
Pull, SAML Attribute Query			X
Push, SAML NameID Management			X

3.1. On-the-fly using SAML during the first sign-on (provisioning)

SAML 2.0 attribute statements are used for provisioning each end user to the service on-the-fly when s/he signs on (WebSSO) to the service for the first time. In the same way, when an end user's attributes change in the IdP, the new attributes are propagated to the service when an end user logs in for the next time.

Pros

+ fully possible and widely used with current SAML2 attribute statements.

Cons

- Requires user activity. An end user needs to be active and log in to the service.

3.2. Manual push (provisioning)

The home organization takes the first step to provision the users to the service. Some out-of-band channel requiring manual intervention is used. For instance, the IdP administrator may provide the SP administrator with an XML or CSV file containing the user entries.

Because a mass provisioning is a manual procedure, it is best suited for mass provisioning.

Pros

+ simply works

Cons

- requires manual intervention

3.3. Automated push, bilateral (provisioning/deprovisioning)

Similar to section 3.2, but now the push is fully automatic and requires no human intervention. For instance, every night the IdP runs a batch job that goes through a pre-configured list of SPs and provides each of them with a list of new end users (provisioning) and/or a list of expired users (deprovisioning).

The operation and its parameters (e.g. protocol/format, attributes, frequency, peer authentication etc) are based on a bilateral agreement with the IdP and (each) SP. The protocol could be, for instance, SPML or a simple CSV file that an IdP uploads to each SP over a secure connection.

Pros

+ simply works

Cons

- if there are several bilateral agreements for provisioning, maintenance may become work-intensive for IdPs and SPs

3.4. Automated push, federation-assisted (provisioning/deprovisioning)

Similar to section 3.3, but now provisioning and deprovisioning is a federation level service that interested SPs can subscribe from the federation operator.

In other words, the identity federation would provide two services:

- Web single-sign on, based on the SAML 2.0 standard

- provisioning and deprovisioning, based on e.g. the SPML standard

The policy of the federation provides framework from both of the two services, and the IdPs may implement any or both of them. Based on their needs, the SPs can subscribe any of the two services; web-SSO for authentication and provisioning/deprovisioning for maintenance of end user's identities. The implementation of the two services can make use of shared elements; for instance, the provisioning/deprovisioning service can use

- the IdP/SP certificates in the SAML2.0 metadata for peer authentication and transport layer encryption,
- the SAML mechanisms available for deciding which attributes should be released to an SP (e.g. SAML 2.0 metadata or the proprietary Shibboleth Attribute Filter Policy)

Pros

+ two separate approached to solve the two distinct problems, still leveraging a shared infrastructure

Cons

- there are no known deployments

3.5. Pull, SAML Attribute Query (deprovisioning)

Deprovisioning could make use of the standard Attribute Query protocol of SAML2.0 (SAML2.0 Core, section 3.3). For instance, every night each interested SP makes an attribute query to the IdPs for each end user. In the query, the SP could use the PersistentID or any other unique identifier of the user, and the IdP's response would cover the fresh attributes for the user, including if the user has expired.

Pros

+ uses standard features of SAML2.0

Cons

- generates unnecessary traffic, because most attributes change relatively slowly

3.6. Pull, SAML NameID Management (deprovisioning)

Deprovisioning could be implemented by a standard IdP-initiated SAML 2.0 NameID Management protocol request (SAML2.0 Core, section 3.6), where the IdP asks the SP to terminate a relationship represented by a given NameID.

Pros

+ uses standard features of SAML2.0

Cons

- the feature is not used very widely