LDAP Facade CLI pilot

Goals and Scope of the Pilot

The pilot aims at providing access to non-web resources via PAM (e.g. for access to sftp, ssh, console) for users by exploiting the existing AAIs, without the need to modify existing software or to obtain user certificates. The resources to which the non-web access is to be provided are considered stateful not only within single session, but also between sessions. A good example of such stateful resources is data. When stored within one user's login session it shall be also available when the user logs in next time. Thus, when the user is identified by some external identity provider, his (global) identity (and attributes) must be mapped to a stable, local resource. This model requires an initial registration/provisioning step, which practically results in setting up a local account.

The users have the possibility to use the unmodified standard client software they used before.

Another requirement is that the user needs to be logged to the same account using different access and authentication methods (e.g. sftp/password, gridFTP/X.509).

The above mentioned requirements shall be fulfilled by LDAP Facade, which allows federated authentication by simulating standard LDAP authentication.

The goals of the pilot are to:

- analyse additional effort for users to access the resources (comparing to traditional access pattern)
- analyse administrative effort
- analyse security aspects of the solution
- propose possible enhancements
- investigate how the solution may be used for guest IdPs
- investigate how the solution may be enhanced to work with multiple attribute authorities, different to the home-IdP

LDAP Facade Architecture

LDAP Facade (LDF) is a solution for access to non-web resources (e.g. available via SSH protocol) based on the idea of FACIUS implemented by Karlsruhe Institute of Technology (KIT). The architecture of the solution is shown in **Figure 1**. The non-standard components are the web application (LDF portal) and the component with LDAP interface, that holds user accounts (LDAP Facade). The LDF portal has two interfaces:

- for administrators: it performs local authentication and allows to administer the service (configuring federations, SP, rules, events, services, groups, roles, users, statistics, etc.)
- for end users: it performs SAML-based authentication (Web Browser SSO profile), used for user registration (provisioning a local account) and configuring access (registering to resources and setting local passwords)

The registered users are stored in an Apache Directory Server that exposes a standard LDAP interface. The Apache Directory Server (similar to any other LDAP server) allows for registering Java procedures that are called when some event occurs. For instance while the user authenticates to a resource, the registered procedure may authenticate the user against his IdP. As the result, the LDAP facade can be used as a local user manager, as well as for authentication and authorization without any modification to existing client and server software. Not only core code can be kept unchanged, also implementing specific plugins is not necessary. On the other side (collaboration with external IdP), the deployment is the same as for any other SAML-based SP.

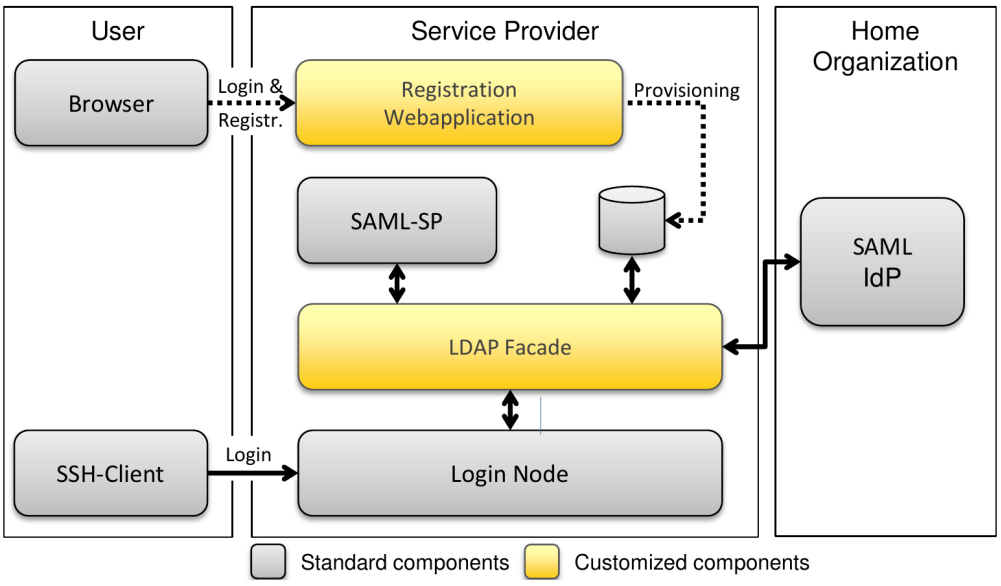

Figure 1. Architecture of LDAP Facade.

source: S.Labitzke Now SAML takes it all: Federation of non Web-based Services in the State of Baden-Württemberg, European Identity & Cloud Conference 2013

Workflow

The solution requires resource provisioning/registration prior to access. This is performed via a web application using the **SAML WebSSO** profile. A local account is provisioned in this step.

The resource requires to have LDAP user management configured, using the LDAP Facade as server. While accessing the resource the mentioned LDAP interface is used for user authentication and as a source of user accounts and groups.

There are theoretically 3 login alternatives to access to the resource:

- **Enchanced Proxy** - requires full trust to the resource server and the LDAP Facade, as they are able to observe the user's password. The client software is not modified, the user sends his password to the LDF SP (via the resource server). The LDF SP authenticates the user and obtains assertions from the IdP using the **SAML ECP** profile.
- **Enchanced Client** - requires modifications on the client side. In this mode, the user has to run a program, that acquires the assertion (using **SAML ECP** or **SAML WebSSO** profile) prior running the client and pass the cooked up password to the client. Recently, a web-page that generates a short lived password for this purpose was integrated into LDF.
  The acquisition of the assertion and access to the resource may also be performed by a single (but non-standard) client.
- **Local Authentication –** the authentication is performed directly to a service-specific password stored in LDAP or via ssh-keys. The validity of the user is checked against his home-IdP using the **SAML Attribute Query** profile.

Alternatives (practically implemented and tested) are described lateron.

Implementation and Deployment of the Pilot

Stable version of the LDAP Facade

This branch of the software (v. 2.5.7 was tested) is used in bwIDM (Federation of non Web-based Services in the State of Baden-Württemberg) service in production mode. The login alternatives in use are Enhanced Proxy and Local Authentication. Piloting this version showed some limitations:

- The service depended on bwIDM specific attribute release, which is not guaranteed to work outside bwIDM.
- SAML ECP and AQ profiles are rarely supported by IdPs.

By the way of work on this version, a solution to login to the resource using pair of cryptographic keys was implemented.

Development version of the LDAP Facade

In order to overcome the above limitations development on LDF is in progress. The tested version introduces "zero attributes" policy, so that the user may login independently on the set of attributes provided, as long as one unique identifier can be found in a set of different attributes (e.g. ePPN, EPUID, ...). For SAML profiles limitation there are two solutions:

- The user logs in (using WebSSO) to an LDF portal and obtains a one-time token. This token may be used as password in the client application (e.g. via copy-paste). The token is verified upon authentication at the resource. This solution is implemented and was tested successfully.
- The user logs in (using WebSSO) to an LDF portal and the account is activated for a short, limited time. The user must login to the resource with credentials local to the LDAP server within this limited time. In particular, when a key-pair (e.g. in ssh) is used for authentication, there is no need to provide password or token. Then this time limit is verified while authentication on the resource. This solution was proposed, but not tested.

Effort and user friendliness

User side:

💡 Supports registration step (accept terms and conditions, setup local password if required) -to be done once via web interface.

👍 Unmodified (standard/legacy) client software

👍 If ECP or AQ SAML profile can be used, the user may login directly to the resource.

👎 If ECP or AQ SAML profile cannot be used, the user must login to the web interface prior logging in to the resource. Then, depending on the solution, the user must pass an obtained token to the client application or login with pair of keys within limited time.

👎 Lack of help/howto.
Administrator side:

👎 Software is not packaged, must be compiled, deployed and configured by the admin

👍 Good installation documentation

👍 No modification to the software on resource side (e.g. standard SSH server can be used), only the proper configuration of authentication and authorization mechanisms must be performed (PAM-LDAP modules)

👍 The web portal is complex -gives lots of functionality (resource management, group management, rules, statistics)

👎 Lack of portal help/howto and general documentation (description of concepts etc.)

👎 There is need for certain versions of underlying software, thus it is recommended to install some pieces manually

👎 The piloting showed some issues with underlying software

👎 Admin interface is not completely translated to English

Security aspects

The solution is correctly designed from the security point of view in general.

- The authentication to the resource is done against user's home IdP, but must be carefully configured (PAM module) otherwise the user may login using local password. (the authentication information is up to date)
- The ECP solution requires to trust the resource provider, as password to IdP is passed through the service, other solutions lack this drawback.
- The user has to register to the resource and confirm terms and conditions (provider's interest is taken into account)
- There is no possibility to lock access to resource for unwanted user by resource/LDF administrator (the admin may deregister the user from the resource, but the user may register again himself).

Demo

The demo environment is available in public. It consists of two elements:

- LDAP Facade portal - SAML SP (https://ldap-facade.aarc-project.psnc.pl)
- Exemplary SSH service (rhus-143.man.poznan.pl)

The suggested workflow follows:

- Prior configuration:
  - User's IdP must be accepted by the SP and vice versa. SP metadata: metadata.xml.
  - Firewall to the resource must be opened for the user.
  - Please contact jankowsk@man.poznan.pl to configure the above.
- Register to the SSH service.
  - Login to the portal and register to the rhus-143.man.poznan.pl service. Your account on that machine will be created automatically.
  - Click on "Registry Info" and check LocalUid.
  - Set (local) password for the service.
- Login to rhus-143.man.poznan.pl with LocalUid as username and password set in the previous step.

Resources

- LDAP Facade documentation: http://wiki.data.kit.edu/index.php/LDAP-Facade
- `LDAP Facade development repository (used in the above demo): https://git.scc.kit.edu/jz9384/reg-app-dev.git`
- `LDAP Facade stable repository (used by bwIDM services): https://git.scc.kit.edu/simon/reg-app.git`