Using PKI to Provide Credential Delegation in non Web-based Federations

Daniel Kouřil, Marcel Poul, Michal Procházka

CESNET z.s.p.o.,
Zikova 4, 160 00 Praha 6, Czech Republic
first.last@cesnet.cz

Abstract. Authentication is basic functionality required by most services that provide access to protected resources or personalized content. In order to authenticate to services users maintain sets of credentials that they use to prove their identity. Credential delegation allows users to seamlessly access multiple services across the network. The concept manifested their utility in the scope of single domain authentication mechanisms. Therefore, emerging identity federations are expected to provide similar functions, too. Recently, various non web-based federation models have emerged, unfortunately they do not cover properly delegation of credentials. In this paper we introduce a mechanism utilizing digital certificates and PKI, which provides support for credential delegation in non web-based federations. The viability of the concept is demonstrated on integration of the mechanism with the Moonshot federation framework. However, the solution forms an independent middleware layer that can be used by several federation models.

1 Introduction

Authentication and access control are two crucial security functions that are needed by virtually all services on the contemporary Internet. Both of them have attracted a lot of attention and plenty of solutions exists that provide support for them. Nevertheless, there are still areas requiring further notice, especially with regards to new technologies that are emerging nowadays. In this paper we focus on delegation of credentials in the scope of identity federations, aiming specifically at non web architectures.

Credential delegation is a mechanism to transfer authentication credentials of a user so that that the transferred credentials can be used to authenticate their owners as if they were the original credentials. A delegated credential can be further delegated to facilitate secure access to services at multiple locations. It is common to restrict the applicability of delegated credentials to limit the impact of abusing credentials that were compromised.

In this paper we focus on the domain of identity federations, especially on non web-oriented architectures that have been emerging. Provisioning of credential delegation is insufficient in federated mechanisms, which either neglect

K.J. Kim (ed.), *Information Science and Applications,*
Lecture Notes in Electrical Engineering 339, DOI 10.1007/978-3-662-46578-3_62

the problem or support just a generic framework lacking a particular delegation mechanism that could be profiled for real deployment.

There are two basic ways how support of credential delegation can be enabled in existing federation mechanisms. A straightforward solution is to provide an ad-hoc support for particular mechanism. It is relatively easy to accomplish, however, such adaptations would have to be done independently for every single mechanism. Another option is to provide a more generic solution that can be easily integrated with existing federated mechanisms.

In this paper we address the lack of delegation support in federated mechanisms by providing a general middleware layer. We decouple the functions for credential delegation from actual authentication mechanisms, which makes them possible to be used independently. The solution forms itself a middleware layer providing delegation, which can be reused with multiple federation frameworks. The viability of the solution is demonstrated on integration with the Moonshot federated protocol suite.

2 Delegation in non web-based federations

This paper aims primarily at identity federations and expects their common arrangement, with service providers providing access to services and identity providers acting as authorities that perform user's authentication and issue assertions claiming user's characteristics. The concept is quite common nowadays and several middleware suites exist nowadays to implement identity federations.

There are a number of scenarios demonstrating the need for credential delegation and SSO in a federated environment. The typical example comes from the area of high-performance computing where user submit batch jobs that need to access large datasets locates at remote storages. In order to retrieve the data the computing jobs needs to authenticate using its owner's credential. Ideally the credentials are delegated during the submission process.

2.1 Delegation tokens

A common mechanism to provide delegation of credentials is to introduce a notion of *delegation tokens* that are used instead of the main credentials. The token must prove the authentication of its owner in the same way as the original credentials but its usage is limited to limit damage caused by its potential abuse. Tokens are used for delegation to another services or users. A token can be created as a result of previous delegation and can be delegated further.

Delegation tokens must provide a few basic functions to be suitable for delegation of credentials. In terms of identity proving, they must provide the same level of assurance as the original credential and should be issued by the same identity provider. Especially tokens cannot be forged without access to the credentials. Delegation tokens should be limited in time and/or scope for which they are permitted to be used. Tokens are created on user's request and another token can be derived from an existing token, providing additional delegation levels to another service. Tokens can be used without user's explicit intervention.

2.2 Delegation in existing mechanisms

Delegation tokens are often provided as part of a particular authentication schema, for instance Kerberos, X.509 proxy certificates, SAML2 and OAuth2, however all of them have some limitations.

The most recognized one from the non-web oriented domain is Kerberos [9], which addresses area of a single administrative domain. Kerberos supports delegation, but the functionality is limited to a single realm that is managed by the authentication instance. A well known example is Microsoft's Active Directory (AD) which is based on Kerberos additionally supporting restricted delegation. Deploying Kerberos in a large scale is practically impossible due to the centralized authentication component (KDC).

In the grid community X.509 proxy certificates [12] are used for delegation, and they are much more suited for a federated environment. Nevertheless, X.509 proxy certificates are not suitable for end users due to complicated usage and limited support in client applications.

Current web-based federations using protocols SAML2 [3] or OAuth2 [6] usually employ time-limited HTTP cookies as authentication tokens, but delegation is supported only by few of them. OAuth2 is currently one of the most often used protocol for delegation, however the delegation can be done only for the first service, subsequent delegation is not supported. In OAuth2 after the successful authentication with the user's identity provider user gets delegation token which is transferred to the end service. Additionally the user can specify what kind of information will be available for the service which receives the delegated token. SAML2 protocol which is widely used in academia has build-in support for delegation, but it is not commonly used due to non trivial deployment and configuration. SAML2 *Enhanced Client Proxy* (ECP) defined in the SAML2 profile is designed for non-web usage of SAML2 together with delegation support, but missing support on most client applications and libraries is the main reason why it is not widely used. ECP version 2.0 has been recently standardized [4] which strengthens security and part of the specification is available in latest versions of middlewares[1]. Both SAML2 and OAuth2 requires initial authentication to be done using web-based tools, therefore support for non-web applications is very limited.

None of them was convenient for the delegation support in the non web domain or they were too complex to integrate. Fortunately, all of the mentioned authentication schemes support transport of additional information from the identity provider to the service, which can be used to integrate delegation support.

In order to address the lack of delegation support we have designed a solution providing generic delegation tokens. The X.509 [5] format was chosen as a suitable mechanism to support delegation. The design of the method is described in the following section.

[1] `https://shibboleth.net/`

3 A PKI-based middleware layer for delegation

To introduce support for X.509 it is necessary to provide a public key infrastructure to manage certificates and their life-cycles. The designed PKI maintains short-lived X.509 certificates that are used for subsequent authentications of clients with their identity providers. Users use the certificates and corresponding keys instead of their long-lived credentials that do not leave their desktops. The creation of a new certificate is governed by the IdP that provides the delegation certificate to the service during standard authentication of the user. Resulting certificates serve as a replacement of the standard, long-lived credentials of users.

X.509 certificates along with the appropriate private keys form delegation tokens fulfilling the requirements. Their management is completely transparent for the users who do not need to be aware of the credentials at all and are not exposed to usual certificate and key management issues. Private keys are not encrypted and can therefore be used transparently by applications.

It is a good security precaution that delegated credentials are restricted in order to prevent from damage caused by stolen or misused delegated credentials. The certificates produced by the PKI are therefore only short-lived, limited to couple of hours of lifetime. Beside the short lifetime, it is also possible to add additional restrictions specifying where the credential is allowed to be used. The user can describe the allowed usage and bound this limitation to the credentials. In our PKI we use X.509v3 critical extensions to convey the limitations, similarly to the concept of X.509 proxy certificates [13]. There are several languages to express the policy, such as ODRL [8] or XACML [1].

To issue X.509 certificates in our concept we use an on-line CA that signs certificates on-demand and is available all the time. Even though the concept breaks some conservative notions of PKI, it is well established in the domain of grid computing and brings several advantages.

In our concept the on-line CA is installed along with the identity provider and provides a mechanism to express the assertion claimed by the IdP in the X.509 format. If requested in the authentication request, upon successful authentication of a user the CA issues certificates to service that the user is authenticating to.

The IdP in turn accepts authentication using these certificates, if presented by the clients as if they were standard, long-lived credentials that are owned by the user. In this way it is possible for a user to delegate their credential to the services.

The design requires that certificate requests and final certificates are transported between the IdP and service. Support for such a transport must be provided as part of the actual protocol that is being extended.

Compared to general-purpose PKIs, our PKI is simpler, which streamlines its deployment and does not increase operations cost. For instance, because of the short lifetimes, the CA do not run any revocation service. Relying party is only the IdP, which also eases operations.

Even though the CA certifies the users in its constituency, it actually does not constitute a new trusted service. From the operations standpoint the CA is a module of the IdP that transforms assertions to the X.509 format. That fact

is important to comprehend since the introduction of the CA does not change the main trust relationships in the federated arrangement as the certificate are only consumed by the IdP itself.

X.509 is widely supported in various middleware and applications, which makes it easier to enable its support in existing application and protocols.

An integration with a particular federation middleware is shown in the next section.

4 Credential delegation for Moonshot

The main goal of the Moonshot project [7,10] is to develop a generic interface for access to end services, using the identity federation approach. The primary focus of Moonshot is aimed at non-web based environments, however its results can be utilized in classic web-based federations, too. The main goal of the project is to design a mechanism that combines the EAP [2], RADIUS [11] protocols and SAML. Although authentication of users is well covered by Moonshot, a suitable delegation mechanism is not supported.

4.1 Moonshot architecture

Moonshot is built upon well known and proven mechanisms for user authentication and authorization. It leverages from well-tested approaches and systems that have been operated for a long time. The first pilots relied on the Eduroam authentication fabric, which provided a scalable hierarchy of authentication servers.

In Moonshot, the EAP protocol is used for the user authentication. From non-web identity federations, Moonshot adopted the user's attribute distribution mechanism based on the SAML. The language is used to carry the assertions about the user made by the IdP to the federated service. Authorization based on the SAML assertions is hence available in Moonshot and application can make use the same attributes that are distributed by web identity providers.

Moonshot benefits from the usage of EAP since the user's credentials can be safely delivered to the authentication server inside the RADIUS protocol. EAP is a widely used protocol that is independent on the underlying transport protocol and supports many authentication mechanisms such as TLS and various password-based methods. Moonshot therefore supports as many authentication mechanisms as EAP does.

The RADIUS protocol serves for the authentication, authorization and accounting. In Moonshot, federated services adopt the role of the RADIUS clients. The home RADIUS server (user's Identity Provider), is chosen upon the user's realm information in RADIUS attributes in the message. Servers are connected in a hierarchical manner so that each of them knows its parent and child. RADIUS protocol can carry different types of messages and when needed it can be extended to convey additional attributes, encoded as *Attribute Value Pairs* (AVPs).

The RADIUS infrastructure and protocol provides the actual federation framework connecting identity providers of multiple institutions. It is used for the credentials delivery from the service to the authentication server as well as the authentication result and user's attribute delivery back to the service.

In the rest of the section we describe how the Moonshot framework was extended to support delegation using PKI.

4.2 Changes to the protocol and profile

We have applied several enhancements on the level of the RADIUS protocol. The integration did not require any significant changes to the core protocol and all adaptations are easily achieved by using custom AVP attributes and appropriate fields of conveyed in SAML messages.

To allow the client to request credential delegation we have introduced two AVP attributes, `Request-Delegation` and `Delegation-Policy`. The former is meant to bear a binary indicator whether delegation should be performed or not, and the latter contains optional delegation policy restricting the usage of the delegated credentials.

In our pilot implementation we have chosen a simple language to enumerate IP addresses. By specifying a list of allowed IP addresses the user can determine the set of machines from which it is allowed to use the delegated credentials. Authentication attempts originating from other machines would be immediately rejected by the RADIUS server. Instead of IP addresses it is also possible to use other kinds of identifications of the servers, namely the authentication service names.

In order to provide a means how a certificate request can be carried from the service to the RADIUS server, we added another AVP attribute called `Certificate-Request`. The attribute is supposed to convey a certificate request in the standard PKCS #10 format.

Final certificates issued by the IdP CA are sent as part of the standard SAML attributes that the RADIUS server encloses to the final reply to the service. The certificate is accommodated in a dedicate SAML field of the response and thus is available as a normal attribute.

Moonshot benefits from the use of the EAP and its authentication mechanisms, which includes TLS support. There were no additional changes needed to the authentication protocol with respect to the usage of the delegated certificate for authentication, simple client and server configuration was sufficient.

4.3 Changes to the Identity Provider

To integrate the PKI with the RADIUS server two crucial steps had to be accomplished. It was necessary to provide the CA support and also make sure that certificates issued by the CA can be used for authentication of clients, subject to possible delegation restrictions.

To provide the function of the CA we provided a set of tools using the OpenSSL command suite that implements manipulations with certificate requests and certificate signing. Using these tools we implemented a module for the FreeRADIUS server that is used to produce X.509 certificates during the authentication exchange. The module is invoked after successful authentication of a client and is passed authenticated username of the client. After it has been invoked, the module constructs an X.509 certificate structure, optionally inserts policy restrictions into the certificate extensions and signs the certificate using the CA signing key. If the client authentication was done using an already delegated certificate, the expiration time of the new certificate is set equally. In other cases, a lifetime of ten hours is used.

While support of authentication using X.509 was trivial, additional steps were required to check the restrictions imposed on the delegation token. Time-related restrictions of the certificate (expiration time) are checked in the standard authentication process but checking the delegation policy had to be done separately. We provided another module that is invoked by FreeRADIUS server during the credential verification step. The module evaluates the policy rules embedded in the certificate and matches them with the IP address of the server that was made available as a standard RADIUS attribute. If the IP address is not allowed in the list, the whole RADIUS request is denied.

4.4 Changes to the client and server sides

To finish the integration we extended the Moonshot implementation to enable client's utilization of the new features.

The server side of Moonshot API routines was extended to expose the delegated credentials to the calling application so that the credentials could be handled properly. We provide an implementation of routines that store the credential on the local filesystem, similarly to how Kerberos tickets or X.509 proxy certificates are stored.

On the client side we added support for utilization of the X.509 credentials if they are found on the local filesystem. Similarly to e.g. Kerberos an environment variable has been introduced (`MOONSHOT_X509_CRED`), which refers to a file keeping the credentials.

5 Conclusion

Both delegation of authentication credentials and support of the Single Sign-On principle are two very crucial functions that users and providers and services require. However, these functions are not provided by all mechanisms. Their lack is especially visible in the field of identity federations, especially non web-based.

In this paper we have introduced a general way how support for restricted delegation can easily be added to a range of existing mechanisms, without changing the way how they do authentication of users and services.

The designed mechanism is based on a public-key schema utilizing the X.509 format. Even though the approach introduces a PKI to transport the delegated information, it actually does not require any changes to the existing trust relationships among the peers.

The viability of the solution was demonstrated by its integration with the Moonshot middleware.

Acknowledgment

This work was supported by program "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005) funded by the Ministry of Education, Youth and Sports of the Czech Republic. Support of the EU GN3+ project is also appreciated.

References

1. eXtensible Access Control Markup Language (XACML) Version 3.0. (2013), http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
2. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H.: Extensible Authentication Protocol (EAP). RFC 3748 (2004), http://www.ietf.org/rfc/rfc3748.txt
3. Cantor, S.: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 (2005), http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf, document ID saml-conformance-2.0-os
4. Cantor, S.: SAML V2.0 Enhanced Client or Proxy Profile Version 2.0 (Aug 2013), http://docs.oasis-open.org/security/saml/Post2.0/saml-ecp/v2.0/cs01/saml-ecp-v2.0-cs01.html
5. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (May 2008)
6. Hardt, D.: The OAuth 2.0 authorization framework. RFC 6749 (Oct 2012)
7. Howlett, J., Hartman, S.: Project Moonshot. Briefing paper for IETF 77, Anaheim (2010), http://www.painless-security.com/wp/wp-content/uploads/2010/03/moonshot-ietf-77-briefing-paper.pdf
8. Iannella, R., Guth, S., Pahler, D., Kasten, A.: ODRL V2.0 Core Model (2012), http://www.w3.org/community/odrl/two/model/
9. Neuman, C., Yu, T., Hartman, S., Raeburn, K.: The Kerberos Network Authentication Service (V5). RFC 4120 (Jul 2005)
10. Painless Security: Project Moonshot: Feasibility Analysis (2010), http://www.painless-security.com/wp/wp-content/uploads/2010/02/moonshot-feasibility-analysis.pdf
11. Rigney, C., Rubens, A., Simpson, W., Willens, S.: Remote Authentication Dial In User Service (RADIUS). RFC 2865 (2000), http://www.ietf.org/rfc/rfc2865.txt
12. Tuecke, S., Welch, V., Engert, D., Pearlman, L., Thompson, M.: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820 (Jun 2004)
13. Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S., Siebenlist, F.: X.509 Proxy Certificates for Dynamic Delegation. In: Proceedings of the 3rd Annual PKI R&D Workshop (2004)