Delegated Authentication Using a SAML Web Browser SSO Profile (v2)

By George B. Dobbs

To comment on this article, please visit our GitHub repository and submit an issue.

Table of Contents

ABSTRACT **2**

INTRODUCTION **2**

TERMINOLOGY **3**

USE CASE **3**

SUMMARY 3
ARCHITECTURE TYPES 3
ACTORS 3
COMPONENTS 3
ASSUMPTIONS 4
PRECONDITIONS 4
POSTCONDITIONS 4
BASIC COURSE OF EVENTS 4
ALTERNATIVE PATHS 5
EXCEPTION PATHS 5
SEQUENCE DIAGRAM 6

AUTHOR BIO **7**

ACKNOWLEDGMENTS **7**

CHANGE LOG **8**

REFERENCES **8**

Abstract

This article builds on a generic IAM reference architecture to describe a common use case of an authentication service using the Security Assertion Markup Language (SAML). We show how a service (the relying party) uses the authentication capability of an identity provider during a web-based single sign-on action.

Introduction

This article is one of a set that illustrates several abstract components defined in the IDPro Body of Knowledge article, "IAM Reference Architecture."[i] This particular article focuses on a specific method of web-based single sign-on via the common use case of a service (the relying party) that uses the authentication capability of an identity provider (IDP) via the Security Assertion Markup Language (SAML) standard. This method allows the RP to delegate the authentication function to the IDP.

This widespread use case relies on trust between the IDP and the relying party (RP). There are various ways of doing this; this article assumes that the trust is based on the use of public-key cryptography, which involves exchanging certificates.

The SAML specification defines three different kinds of assertion statements; this article is only about the authentication assertion.

The SAML specification supports the mapping of identities between different names, known as federated identity. This article is restricted to a single domain, such as an organization providing access for its employees to web-based services provided by third-party vendors. In other words, a single domain of administration allows for the user identifiers to be shared.

Even a cursory review of the OASIS SAML standards documents will reveal an extremely rich and flexible structure.[ii] This article represents a very thin slice of its possibilities focusing on the run-time aspect of authentication using the web (HTTPS) messaging protocol. Technically, we are discussing what OASIS calls the Web Browser SSO profile, using the POST binding.[iii]

This synopsis stresses the importance of the Trust Root. The messaging between the IDP and RP passes through the User Agent. The User Agent must be considered untrusted, as a corrupted agent could potentially modify the messages. To protect against this modification, the messages are protected by a digital signature, which must be validated. It is the common certificate authority that acts as the Trust Root to support these signatures.

The topic of signatures becomes quite deep quickly and is not covered in detail here. The SAML specification relies on the W3C Recommendation XML Signature Syntax and Processing, which may be of interest.[iv]

Terminology

Please see also the terminology in the IDPro Body of Knowledge article, "IAM Reference Architecture."

Item	Definition
User Agent	A user agent is any software that retrieves, renders, and facilitates end-user interaction with Web content. [v]

Use Case

Summary

The web user works through a user agent to access resources at an RP. The access request results in a redirection of the user to an IDP as part of an authentication action. This result of the authentication is an authentication assertion that is consumed by the RP and used to establish a security context for the web user. In effect, the RP has delegated the authentication to the IDP.

Architecture Types

Different architecture types are defined in the "Introduction to IAM Architecture" article in the IDPro Body of Knowledge. The SAML-based authentication use case applies specifically to the architecture of Cloud Environments.[vi]

Actors

The user is the only actor. The user acts through the User Agent (the browser). The other participants in the use-case are systems "actors", which we show as components.

Components

The following components are defined in the article, "IAM Reference Architecture."[vii]

- Audit Repository
- AuthN / Assertion (part of IDP)
- Identity Register
- Relying Party (RP)
- Trust Root

Please note that the SAML documents refer to the relying party as the service provider.

Assumptions

The user wishes to access a protected resource and has requested access via a web browser, the User Agent.

The RP has a single IDP. The RP may support several IDPs, so a method to determine which one to use would be needed in that case.

Preconditions

There must be an established trust between the RP and the IDP before SAML can be used for authentication. "The primary mechanism is for the relying party and asserting party to have a pre-existing trust relationship which typically relies on a Public Key Infrastructure (PKI). While SAML does not mandate using a PKI, it is recommended."[viii]

The IDP and the RP use the same user identifiers. The SAML specification establishes ways to map these, but we don't discuss this subject here.

Postconditions

The user is logged into the RP's site.

Basic Course of Events

The following shows the "happy path", without errors. See also the sequence diagram below. See Alternative Paths for some variations.

1. The user selects the login function on the RP's site. This selection may be automatic when the user attempts to access the protected resource.
2. RP determines that the user is not logged in.
3. The RP prepares an Authentication Request message, which the RP may sign. It is delivered to the user agent as a form targeted at the IDP, which is known since there is a single configured IDP. The user agent (automatically via a client-side script) sends the request to the IDP.
4. The IDP ensures the signing certificate from the RP is still valid by checking for revocation.
5. The IDP validates the request and interprets its contents. The signature and some field values (such as Issuer, AuthnContextClassRef, etc.) are checked.
6. The IDP interacts with the user agent to gather the user's identifier and credentials. For example, this could ask for a username and password, but it could be something else.
7. The IDP uses its Identity Register to validate the credentials.
8. The IDP prepares a Response message, which the IDP signs.

9. The response is then sent back to the user agent with instructions to use an HTTP POST to forward it to the RP. (The target RP URL is typically known to the IDP through initial configuration).
10. The RP ensures the signing certificate from the IDP is still valid by checking for revocation.
11. The RP validates the response and interprets its contents. The signature **must** be checked. The RP checks it against the already active assertions to prevent replay and makes other checks. The RP then determines whether the authentication was successful.
12. Not shown in the chart are the audit records being written. The various components should write these.

Alternative Paths

Step 3 may be replaced with an HTTP redirect. This formulation is an allowed composition of the POST binding and the Redirect binding.[ix]

There are also alternatives to the POST method in step 3.[x]

In SAML terms, this is the service provider-initiated variant. There is also an IDP-initiated alternative.

The messages may be encrypted. For instance, in step 8, the IDP may encrypt, and in step 10, the RP would need to decrypt the response.

Not recommended: some implementations have ignored request signing and signature verification, possibly due to historical performance issues.

Exception Paths

Failure to authenticate at the IDP does not return an assertion.

Failure to validate the signature indicates that the assertion should not be honored.

Various error conditions, such as the validity period expiring, are described in the standard.

Sequence Diagram

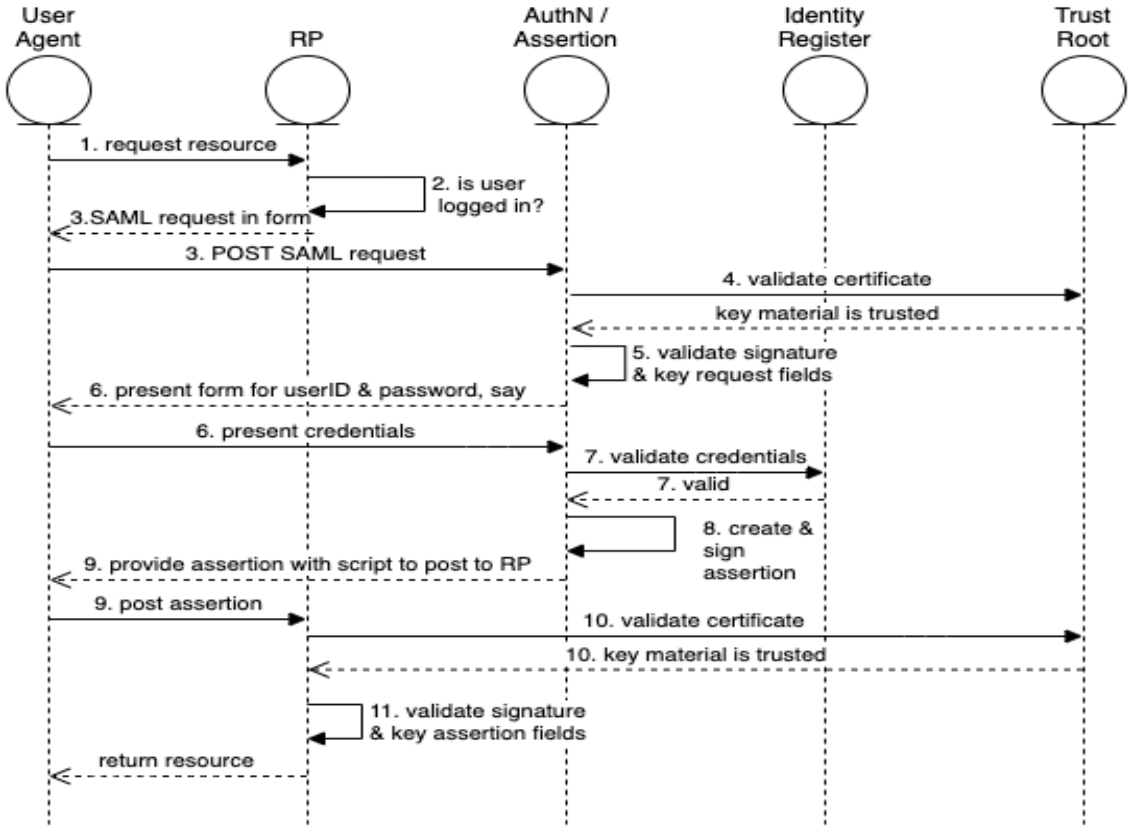

Figure 1: The "happy path" of the Web Browser SSO Profile

Author Bio

George Dobbs manages architects at a major insurance company. He is also the chairman of the IDPro Body of Knowledge Committee. One of his interests is modernizing the use of Identity and Access Management techniques used by the firm. He is particularly interested in the area of customer-facing applications, including approaches to fraud prevention in call center and digital contexts. Related to this, he is interested in the evolution of distributed session management – notably distributed session termination. He is a founding member of IDPro and represented his firm in the Identity Ecosystem Steering Group (IDESG). Prior to his current position, he led the development of customer-facing identity for websites at three other insurers. He has led a local identity and access management user group since 2004. Prior to that, he was the chairman of the Network Applications Consortium.

Acknowledgments

The author would like to express gratitude to Chris Olsen and Bernard Carlier for their review and suggestions for improving this text.

Change Log

Date	Change
2021-09-30	V1 published
2022-12-15	V2 published; title changed, intro and use case summary clarified; Alternate Path includes a “not recommended”

References

[i] Dobbs, George, “IAM Reference Architecture,” IDPro Body of Knowledge, 30 September 2021, https://bok.idpro.org/article/id/76/.

[ii] “OASIS SAML Wiki – Front Page,” wiki page, OASIS, https://wiki.oasis-open.org/security/FrontPage#SAML_V2.0_Standard (accessed 28 November 2022).

[iii] Hughes, John, and Scott Cantor, Jeff Hodges, Frederick Hirsch, Prateek Mishra, Rob Philpott, Eve Maler, eds. “Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS, 15 March 2005, https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf (accessed 28 November 2022).

[iv] Bartel, Mark, and John Boyer, Barb Fox, Brian LaMacchia, Ed Simon, “XML Signature Syntax and Processing Version 1.1,” Section: Core Validation, World Wide Web Consortium, 11 April 2013, https://www.w3.org/TR/xmldsig-core/#sec-CoreValidation, (accessed 28 November 2022).

[v] “User Agent Accessibility Guidelines (UAAG) 2.0,” W3C Working Group Note, https://www.w3.org/TR/UAAG20/#glossary (accessed 28 November 2022).

[vi] Cameron, Andrew, and Graham Williamson, “Introduction to IAM Architecture,” IDPro Body of Knowledge, 17 June 2020, https://bok.idpro.org/article/id/38/ (accessed 28 November 2022).

[vii] “IAM Reference Architecture,” https://bok.idpro.org/article/id/76/.

[viii] Ragouzis, Nick, and John Hughes, Rob Philpott, Eve Maler, Paul Madsen, Tom Scavo, “Security Assertion Markup Language (SAML) V2.0 Technical Overview - Committee Draft 02,” OASIS, 25 March 2008, https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf (accessed 28 November 2022)

[ix] Cantor, Scott, and Frederick Hirsch, John Kemp, Rob Philpott, Eve Maler, eds. “Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS, 15 March 2005, https://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf (accessed 28 November 2022).

[x] Ibid.