Integrated Security Context Management of Web Components and Services in Federated Identity Environments

Apurva Kumar

IBM India Research Lab. 4, Block C Vasant Kunj Institutional Area,
New Delhi, India-110070
kapurva@in.ibm.com

Abstract. The problem of providing unified web security management in an environment with multiple autonomous security domains is considered. Security vendors provide separate security management solutions for cross-domain browser based and web service based interactions. This is partly due to the fact that different web standards dominate in each space. E.g. Security Assertion Markup Language (SAML) which is an important standard in cross domain single sign on (SSO) specializes in browser based access while WS-* standards focus on security needs of web services. However, cross domain web services are often invoked in context of a secure browser session. Considering these interactions in isolation will lead to a fractured security solution. This paper proposes a solution that provides seamless transfer of security context across various types of cross-domain web interactions.

1 Introduction

Web is increasingly becoming the dominant channel for customer interaction with service providers. Providers compete with each other to enable the widest range of services on their websites. However, very few providers are diverse enough to be able to satisfy a wide range of services without interacting with their partners. The interaction can either be an explicit redirection to a partner or it could involve implicit invocation of web interfaces exposed by the partner. Needless to say, such interactions should be both secure as well as trusted by partnering organizations. One of the fundamental problems in designing such security solutions is the lack of a common source of identity information.

Federated identity management solutions address the problem by having authentication and attribute authorities that are trusted by all partners. Such solutions provide secure token exchange mechanisms to convey assertions about authenticated identity to multiple service providers.

At present, there are two standardization efforts in this space. Security Assertion Markup Language (SAML) [1,2] is an important and widely used federated identity management standard from OASIS Security Services Technical Committee. The single most important problem that SAML tries to solve is the Web Browser Single Sign-On (SSO) problem. A set of WS standards (WS-Security [3], WS-Trust [4],

A. Bouguettaya, I. Krueger, and T. Margaria (Eds.): ICSOC 2008, LNCS 5364, pp. 565–571, 2008.

WS-SecureConversation [5] and WS-Federation) collectively address the problem in web services domain. Since the two solutions address problems from seemingly different domains: machine to machine interactions and browser based interactions it might seem reasonable that organizations use both of them independently depending on the type of interaction.

However, very often, interactions between machines are driven by a human action of clicking a link or button on a website. If such interactions pass through organization boundaries, it is often important to propagate the security context. If the security context of the browser interaction is not passed to a web service invoked at a partner organization, it will lead to inferior solutions that compromise privacy or result in unnecessarily increased trust level between partners. Such factors will certainly limit the capacity of web as a medium for carrying out secure business transactions. In this paper we take an example of a telecom service provider to illustrate how existing solutions do not provide satisfactory solution for the problem. We propose a solution that extends the SAML browser based SSO use case to incorporate additional requirement arising out of the more complex interactions required in the trust model. The solution is based on introducing a new type of assertion called a '*Resource Request Assertion* (*RRA*)'.

2 Case Study of a Telecom Service Provider

2.1 Problem Description

Consider a telecom service provider (denoted as SP) having a customer portal where it allows its customers to purchase content (e.g. ring tones, wallpapers, music etc). SP does not host its own content but depends on a content provider (CP). CP supports advanced algorithms for rating of content based on subscriber usage. The agreed revenue model is that CP will charge SP based on the usage profile of the customer, charging less for a heavy (frequent) user and more for a light user. Communication between the two organizations is through two web services exposed by the CP for: *browsing/rating* service and *purchase* service.

When a customer logs on to the SP portal and wants to view a page of contents before purchase, the browsing/rating CP web service is invoked and the customer identity is passed as a parameter. The prices displayed to the customer are set by SP based on rating information received from CP for the customer. As customer continues browsing this interface is invoked multiple times by the SP portal application. The customer then chooses to purchase a content in response to which SP charges the customer and sends a request to the purchase interface of the CP web service. In response, CP provides a download URL to SP. The download URL is in CP domain.

To manage these transactions securely, the two parties approach another service provider (IDP) which provides identity management solutions. They agree on trusting IDP for authentication and as an authority for issuing, validating and exchanging tokens. The IDP sets up a customer repository which is maintained synchronized with the customer master (e.g. a CRM database) of SP. The identity provider website also provides web registration facility to customers of SP.

For CP, its content is the key and it wants to ensure that SP should not be in a position to take advantage of the charging model. For SP, its subscriber base is the key asset and it wants to ensure that its subscriber details are not misused or divulged to other parties. We now consider some solutions based on available federated identity management technologies.

In the following discussions, we assume there are four types of links on the SP website. *GUEST* links are the only ones which can be browsed without sign in. *BROWSE* links are those that require access to browse/rating web service from CP. *PURCHASE* links are those which require access purchase web service from CP. *DOWNLOAD* links are those which are redirected to CP website for downloading purchased content.

2.1.1 Solution Approach 1: SP Asserts Customer Identity

IDP proposes the following first solution in which it handles browser and web service interactions using a uniform approach, but independent of each other. For browser based interaction, the federated identity is that of the end user. For web service based transactions, the authenticated identity is an application ID, which identifies an SP application that invokes the web service exposed by the CP.

Figure 1 shows the steps executed in a typical user session in which user browses and then selects a content to purchase and download. These steps are described below:

Step 1: Customer connects to the SP website and browses.

Step 2: On clicking at a BROWSE link, the browser is redirected to the IDP website. An authentication request is encoded in the redirection URL.

Step 3: The IDP site throws a password challenge page to the customer.

Step 4: The user credentials are validated by the IDP and a token is issued. The token is digitally signed by the IDP. Also a security context is created for the user. The token is embedded in an HTML page returned to the browser (e.g. as a hidden form control).

Step 5: An auto-submit script causes the token to be HTTP POSTed to a URL in SP domain which is a consumer of assertions provided by the IDP.

Step 6: The token is validated and a new security context is created at SP for the customer and the customer is logged in. The originally requested URL is retrieved and forwarded to the SP application. In processing the BROWSE request, the SP application needs to invoke the CP browsing/rating web service. Since all customer requests are routed through the same application, the application is already signed in with the IDP and shares a security context with the web service (e.g. through a WS-SecureConversation [5] secure context token, SCT). The CP web service is invoked and the customer identity is passed as a parameter.

Step 7: CP confirms the security context is valid and then processes the request. The customer identity is used to provide rating based on customer usage. SP uses the result of the web service, maps rating points to prices and displays the catalogue/content to the customer. Steps 6-7 might be repeated multiple times, till the user decides to buy an item.

Step 8: The customer selects a content item and clicks on PURCHASE link. The SP web application initiates a charging request for the customer (e.g. by calling a web service in its own domain). After successful charging, SP invokes the purchase web service of the CP. The security context (e.g. based on an SCT) used in Step 6 is used.

Step 9: The CP returns the download URL in its own domain, which is displayed as a link on the SP website to the customer.

Step 10: The customer follows the link and is redirected to the CP website.

Steps 11-14: The sign on steps 1-4 are repeated for CP. However, this time authentication of customer is not required, since the browser already has a security context with the IDP. Once the token is verified by CP, it allows the user to access the content.

Summary. In an SAML based solution, steps 1-5 correspond to SAML Browser SSO profile [2]. Steps 6-7 correspond to accessing the browsing/rating web service. Step 8-9 corresponds to accessing the purchase web service. Step 10-14 correspond to downloading the content after the SAML browser SSO profile is repeated with the CP.

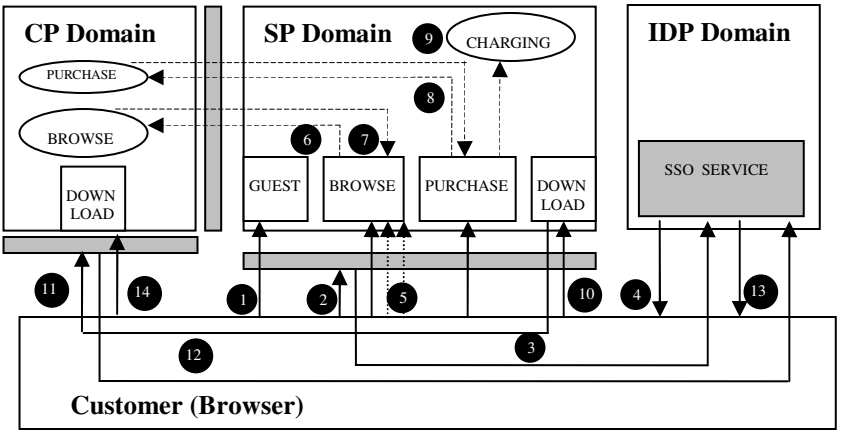


Fig. 1. Case Study: Sequence of events in solution approach 1

Analysis of Trust Model. The customer identity used by IDP can be a pseudonym rather than an identifier relevant to the business, thus the approach does not risk privacy of SP customer data. However, the trust model does not work quite so well for the CP. In Steps 6 and 8, it has to trust SP assertion about the identity of the customer. The revenue model is based on both the volume of each content item purchased as well as the purchaser. In this model, for the same sequence of contents downloaded the revenue for CP will be more if the content is accessed by light users as opposed to frequent users. If CP has to trust identity supplied by SP, it is possible for SP to replace light users by heavy users while asserting identity, thus bringing down the cost to be paid to CP.

2.2 Solution Approach 2: CP Controls Content Delivery

To address the above problem, IDP proposes an alternative approach in which CP controls delivery of content by sending it directly to the customer. This solution goes through the following flow:

Steps 1-8: Same as approach 1.

Step 9: CP returns the download URL (which is in its own domain) to SP. It also associates the URL with the pseudonym in the request.

Step 10: Customer accesses the DOWNLOAD link on SP website and is redirected to the CP website.

Steps 11-14: The sign on steps 1-4 are repeated for CP. Same as approach 1.

Step 15: CP sends an attribute request to the IDP with the pseudonym corresponding to the accessed URL for retrieving mobile number of the customer.

Step 16: The user is asked to confirm the mobile number for which the content was requested. Once user confirms, the content is delivered directly to the mobile device.

Analysis of Trust Model. This approach meets requirements for the CP since it is able to ensure that the content is delivered to a mobile number of the user asserted by the SP. However, this approach requires that CP has access to mobile numbers of SP subscribers. Since CP already has the usage profile for the customers, this knowledge allows CP to target subscribers of SPs network through other channels.

2.3 Need for Resource Request Assertion

Both the solutions described though feasible do not satisfy all the requirements of collaborating parties. The basic issue is that when a web service is called in the context of a browser SSO session, there is no secure means of passing the identity information from browser to the service. The problem arises because the access from SP to CP is direct without involvement of the IDP or the browser.

As a trusted party, it would have been ideal if IDP had certified that the browse/purchase URL for which the web service has been invoked was accessed by the customer. We call a statement binding an authenticated subject with a resource as a *Resource Request Assertion (RRA)*. However, this type of assertion is not included in the assertions types available in major federated identity management standards. E.g. SAML supports authentication, authorization and attribute assertions.

We now outline strategy for solving the integrated browser SSO and web service security problem. First, we should use extensibility of XML based federated identity standards to define a new assertion type: resource request assertion. Next, we should incorporate request and response messages for the new token in the browser based SSO flow. Finally, we use this token to propagate browser security context to the web service. In the following section we propose a solution based on this strategy.

3 Proposed Approach for Integrated Web Security Context Management

In the proposed approach (Figure 2), we use a Resource Request Assertion (RRA).

Step 1-7: Identical to approach 1. The BROWSE links are accessed as before.

Step 8: Customer selects a content item and clicks on PURCHASE link.

Step 9: The browser is redirected to the IDP website with a request for an RRA token. The requested link can be passed in the name field of the subject element in an SAML exchange.

Step 10: IDP verifies that a login context for the user exists. It then retrieves the requested URL and presents a page to the customer to confirm that he has requested access to the URL.

Step 11: User confirms the access request. IDP issues a signed RRA token which binds the URL with the authenticated subject (user). The token also contains the time of the request as well as validity period. It embeds the token in an HTML FORM control and returns the form to the browser.

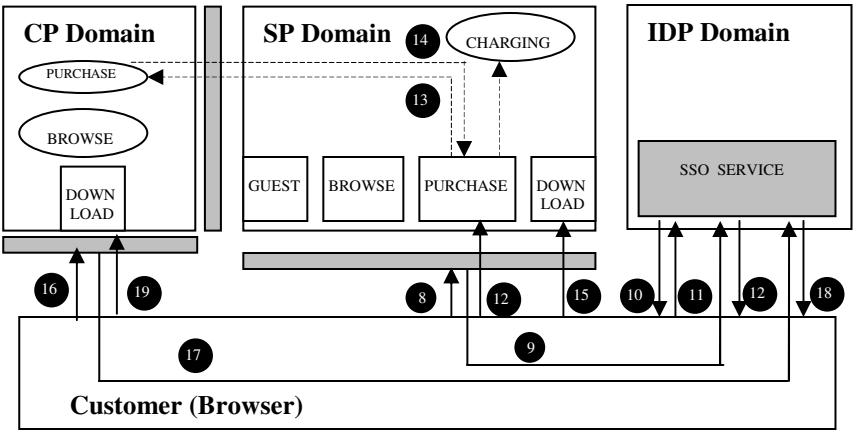
Step 12: An auto-submit script is executed on the browser which POSTs the form to the assertion consumer service of SP.

Step 13: The assertion consumer service forwards the request to the SP application after inserting the RRA token as an HTTP header. During processing of the request, the SP application needs to invoke the CP purchase web service to get the content URL. As in solution 1, we assume that the web application is already logged in to use the web service through IDP. The SP web application initiates a charging request for the customer (e.g. by calling a web service in its own domain). Finally the purchase interface of web service is invoked. The RRA token in the HTTP header of the request is used to obtain a new security context based on the existing one. Details are omitted due to space limitation, but this is done using WS-Trust token exchange facility.

Step 14: CP associates the new security context with the user (customer) identifier in the RRA token. It confirms that the user identifier passed in the purchase request matches with that in the token. *This is the key step instrumental in solving the trust problem of solution approach 1*. After this verification the request is allowed to proceed as earlier. Finally the download URL of the content is returned as a result of the call.

Step 15-19: Identical to Steps 10-14 in solution approach 1. The user clicks the download URL to be redirected to the SP DOWNLOAD page. The normal SAML browser SSO profile is executed between CP and IDP. Finally, the user is allowed to download the content.

Summary. In an SAML based solution, Steps 1-5 correspond to SAML Browser SSO profile for signing on SP website. Steps 6-7 correspond to accessing the browsing/rating web service. Step 8-12 corresponds to issue of an RRA token. This exchange is very similar to the browser SSO exchange. Steps 13-14 correspond to invoking the purchase web service at the CP. Steps 15-19 correspond to downloading the content after the SAML browser SSO profile is repeated with the CP.



.

Fig. 2. Case Study: Sequence of events in proposed solution for integrating Web Security Context Management

4 Conclusion

We demonstrate by means of a realistic case study, the problems that can arise in security solutions which ignore links between browser based and machine to machine communications. We propose a new type of assertion, called the Resource Request Assertion (RRA), which binds a subject with one or more requested resources (e.g. an HTTP URL). We use RRA as the means of controlling security context of a web service interaction on the basis of the browser security context. We use SAML as the browser SSO protocol and WS-SecureConversation and WS-Trust for web services security context management to provide a concrete framework for implementation of the solution. The RRA concept is powerful and it should be possible to use it to define other trust models not necessarily restricted to the federated identity domain.

References

1. Cantor, S., et al.: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
2. Hughes, J., et al.: Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
3. Nadalin, A., et al.: Web Services Security: SOAP Message Security 1.0, WS-Security 2004 (2004), http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
4. Anderson, S., et al.: Web Services Trust Language (WS-Trust) (February 2005), http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-trust/ws-trust.pdf
5. Anderson, S., et al.: Web Services Secure Conversation Language (WS-SecureConversation) (February 2005), http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-secon/ws-secureconversation.pdf