SAML

Technical Specifications:

Programming: C#, ASP.Net MVC 5
Database: SQL Server 2014
Third party Tools: Component space SDK for SAML in MVC
Design : HTML5, CSS3, Bootstrap, jQuery
Hosting: RDC server

Background :

The key focus of this project is to implement SAML based SSO mechanism using "Component Space" in an existing .Net based application. This mechanism would transfer the user's identity from one place (the identity provider) to another (the service provider) through an exchange of digitally signed XML documents. The application URL would be provided within the end-client application where the SAML is used and the mechanism would pass token to automatically open the application.

Salient Features of SAML :

Service Provider (SP): Service provider is the one, which hosts applications.

Identity Provider (IDP): An identity provider is a trusted provider that enables you to use single sign-on to access other websites.

Why SAML is needed?

One Secure SSO Portal for All Apps

Single sign on allows users to access multiple services with single login. By using Single sign-on, user no need to remember number of usernames and passwords. Prior to SAML, products support single sign on by using browser cookies. User authentication state information is maintained in browser cookies, so that re-authentication is not required each time the web user accesses the system. One problem with cookies is, cookies are not transmitted between different domains. With this mechanism users only have to enter one set of credentials to access to their web apps. This greatly increases productivity while keeping data secure. It enables password security and multi-factor authentication ensuring that only authorized users get access to sensitive data.

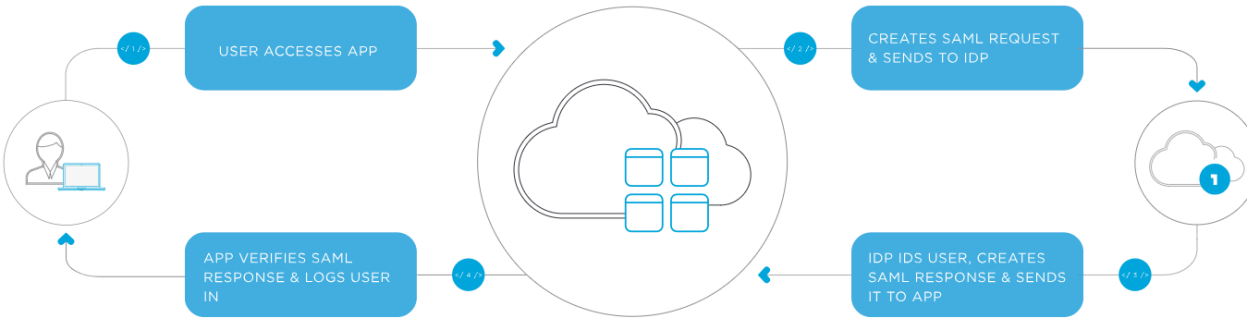
How SAML Works :

The sequence of events goes like this:

When a user tries to access the service provider, the service provider in turn checks to see if the user is already authenticated within the system. If not, the service provider starts the authentication process. The service provider redirects the user to the single sign-on (SSO) service.
User's browser sends an authentication request to the SSO service; the service then identifies the user. The SSO service returns an XHTML document, which includes the authentication information needed by the service provider in a SAML Response parameter. The SAML Response parameter is passed on to the service provider. The service provider processes that response and creates a security context for the user; basically, it logs the user in and then tells him where his requested resource is. With this information, the user can then request the resource he is interested in again. The resource is finally returned to the user.

SAML SSO Flow:

The diagram below illustrates the single sign-on flow for service provider-initiated SSO, i.e. when an application **triggers SSO.**



Component space SDK :

Component space SDK succor the integration of SAML single sign-on with easy and comprehensible APIs. It provides a seamless, secure access to cloud and corporate web applications using a single username and password.

Challenges/Problems :

The main challenge was the implementation of single Sign Out in an existing website. The application had multifarious applications attached to it. The client wanted us to implement a mechanism that would allow the users to login once and gain access to all systems they have authorization to use without any additional login prompts .

Vulnerability Problem:

However, there are also other risks involved with this mechanism. Using the same password on all your various web applications could be treacherous to let one username/password combination unlock all the resources an individual user has access to.

Problems with Received SSO method using IdP :
Method using Identity Provider Initiated SAML and High Level APIs.

Attribute not properly Mapped :

The attribute containing the user-name is not properly mapped as specified in the Remote User ID field in the Map SAML Attributes section on the SAML Authentication Settings page.

Solutions :

- The corollary of this mechanism is indeed the improved user experience through automatic login. No matter how challenging it is to develop this mechanism, it always feel revering when it saves time and effort .

- To avoid vulnerability issue, we stipulated stronger passwords, since the need for multiple passwords and change synchronization is avoided.

- After a thorough understanding of the client's vision and requirements, we came up with a solution of adding an event will be logged in the bb-services log when attempting to login.

Results:

- With SAML, we need not to maintain multiple admin sections in multiple projects. The identity provider bears this burden.

- Refined User Experience: This mechanism provides improved user experience.

- **Resource savings**: Reduced IT help desk costs, by reducing the number of calls to the help desk about lost password.

- **Security:** The Users credentials are provided directly to the central SSO server, not the actual service that the user is trying to access, and therefore the credentials cannot be cached by the service.