



December 2004

<http://www.intelligententerprise.com/showArticle.jhtml?articleID=54200324>

# SAML: The Secret to Centralized Identity Management

**Complicated by too many systems, too many applications, and too many passwords, identity management is a major headache for most organizations. Can an intelligent, Web-services approach employing new standards ride to the rescue?**

By Hank Simon

The last time some of our internal users and external partners counted, they had more than 15 passwords they had to keep track of. Of course, they could keep all those 15 passwords in their heads. Yeah, right! Every time they needed new access to a new resource, application, or data set, they had to find the responsible administrators. And the administrators were always available, never on vacation, and always had a backup admin. Yeah, right! And when users left the company or were terminated, or when partners became competitors, the administrators were always informed so that they could disable access. Yeah, right! And in this dream world, we know that the CIOs were happy and always received compliments from the user community on the ease of getting to the data. Yeah, double right!

The term that covers many of these issues is called identity management, and the CIO asked my team to look into the situation to see if we could improve it.

Identity management refers to provisioning, password management, and access control. Typically, access rights are stored in different locations, with separate access-control lists for individual applications and resources. Identity management must control data, people, and resources that are distributed across different locations. Historically, a multitude of separate systems handle identity management functions. For example, one program handles provisioning, another manages passwords, LDAP stores authentication information, and each application (or administrator) maintains individual user access-control lists. Keeping these separate functions maintained, synchronized, and up to date is a resource-intensive, costly proposition.

We've developed an authentication/authorization (AA) Web service that unifies the functions of identity management as a first step toward the goal of a federated, enterprisewide, single sign-on solution that improves our identity management problem. This AA Web service is a multistage set of services that enables a user to login once and then invoke a Web service to access applications and data resources. This approach has two major benefits. First, it assists in the simplification of identity management by transferring access-control management from multiple, local applications to a centralized authority, such as LDAP. Second, this approach provides a general method to allow Web services to gain access to data.

## Centralization is Key

We used LDAP as our centralized authority by adopting a rules engine by Jericho Systems, a software solutions company that has developed a security package called EnterSpace. EnterSpace includes a SAML

service and the rules engine as one component of the security package. We selected Jericho Systems because the company offered a good product, a good price, and it was willing to work with our constraints. An advantage of EnterSpace is that you can use SAML with the rules engine and customizable rules to link all components of identity management to LDAP as a centralized authority.

Centralizing and linking provisioning, password management, and access control makes life simpler. It's natural to link identity management processes to LDAP, as a reference point and central authority. However, LDAP is typically used for authentication, not for authorization. SAML facilitates our ability to link LDAP authentication with access authorization.

## SAML's Role

In November 2002, the Organization for Advancement of Structured Information Standards (OASIS) ratified SAML as the eXtensible Markup Language (XML) framework for exchanging authentication and authorization information among business partners, particularly through Web services. SAML enables Web-based security interoperability functions, such as single sign-on, across sites that are hosted by multiple companies.

SAML supports secure interchange of authentication and authorization information by leveraging the core Web services standards of XML, Simple Object Access Protocol (SOAP), and Transport Layer Security (TLS). Many vendors, such as RSA, Netegrity, IBM, Oracle, BEA, Oblix, and Jericho have committed to SAML and are implementing the specification in their products.

A SAML assertion uses the header in a SOAP message to pass through HTTP, transferring security information between an assertion authority and a relaying party.

For example, a user can login at one site; a SAML assertion transfers the user authentication token; and the transferred token provides authentication to a remote site. A SAML package can include the authentication token as well as user attributes that can be tested against the rules engine for authorization and access control.

It's important to note that SAML doesn't perform the authentication; rather, it transports the authentication information. In addition, SAML can use different authentication authorities, such as LDAP, Active Directory, and Radius, allowing for different identification methods such as password, biometric, Public Key Infrastructure (PKI), Secure Socket Layer (SSL), Kerberos, and so on. Then, as the transport, SAML passes the assertion information that the user is authenticated. In contrast, SAML doesn't perform authorization or transport access-control information.

Picture SAML as a door with a bouncer, as you might see in the movies. If some shady figure comes up and says "Joe sent me," that means that Joe authenticated the speaker and the bouncer can direct the person to the local poker game. If a glamorous actress appears and says "Rudolph sent me," the bouncer reviews Rudolph's access list of VIPs, locates her name, and then escorts her to see the show. Finally, if James Bond flashes a card and looks into a scanner, he may be authenticated by the card and a retina scan. Then the authentication information is compared to an authorization list that identifies agents who are allowed to enter the secret elevator to see Q.

## SAML Security Risks

SAML has three well-understood potential security attacks:

1. **Replay attack**, which occurs when a malicious hacker hijacks a SAML token and replays it to gain

illicit access

2. **DNS spoofing**, which occurs when a hacker intercepts a SAML token and sends a false DNS address
3. **HTTP Referrer Attack**, which occurs when a hacker tries to reuse an HTTP referrer tag.

Using a timed session can reduce or eliminate the threat of these three attacks. Eliminating the attacks is done by using a token only once and logging the use so that reuse is flagged; by using an IP address to avoid DNS spoofing; and by using HTTPS with SSL/TLS to eliminate an HTTP attack. Experts and analysts agree that these risks can be mitigated and that SAML offers a secure standard for assertion.

## Page 2

### Process Overview

Our AA Web service carries out a number of security operations in sequence. A user is uniquely identified and authenticated by using a username and password pair that is compared to information on an LDAP server. A SAML token, carrying authentication information and user information, is created for each login session. A user is authorized to have access to a data resource by using a rule, which defines access control, as evaluated by the Jericho Systems EnterSpace rules engine.

This general flow is shown in Figure 1. Evaluation includes parsing information from the SAML token. The rules are created based on policy. Because the rules control access and embody policy, the rules engine enforces policy management by implementation. Very generally, a user gains access based on LDAP information, user information, and the access rules. In addition, when users leave the company, all of their accounts are managed with LDAP. When LDAP accounts terminate, then all accesses terminate at the same time. In this way, you can centralize identity management based on LDAP as the centralized authority.

To illustrate, let's look at the steps shown in Figure 1. The user logs into the client on her desktop, which authenticates her via the local LDAP (not shown). The client then sends a request that includes the returned authentication information to the SAML service, which wraps the information for use by the Web service. The Web service verifies the authentication information in the SAML token, by sending a query to the LDAP server (not shown). Upon verification, the Web service allows access to the data resource based on rules in the rules engine (not shown). The Web service sends the response back to the client, for fulfillment of the user's request.



**FIGURE 1:** Sequence of security operations.

### How It Works

The AA service is a complex Web service that adheres to service-oriented architecture (SOA) principles. It's based on SOAP over HTTPS, both of which are open, nonproprietary standards independent of software languages or vendor systems. They offer no costs for use, licensing, or maintenance. Using SOA principles for design means you can construct this Web service in a modular fashion, allowing for removal and replacement as necessary for use and maintenance. Therefore, you can make modifications to any portion of the architecture without modifying other portions.

The AA service uses the local domain LDAP for authentication and subsequent SAML-based security

assertions. The SAML token is used to pass assertions to trusted resources, which then approve or deny access to data resources.

The function is similar to a driver's license, where the Department of Motor Vehicles (DMV) is an authoritative source (like an LDAP) that validates your identity and ability to drive a car. The police officer serves as a trusted resource who can check back with the DMV for identity and then affirm or deny driving privileges to a specific driver based on current eligibility status or rules of the road.

The AA service uses a rules engine to evaluate authorization rules based on security policies that specify conditions for approval or denial of access to data resources. The rules engine is central to reinforcing policy management.

The use of LDAP as the authoritative source also supports a centralized source for identity management. Rather than managing resource access at the individual resources, user access lists can be managed by using the LDAP and the appropriate policies that define user roles and corresponding data parameters, such as department and management level.

Using Figure 2 as a visual key to the architecture, here are the major steps in the process:



**FIGURE 2:** Schematic showing the key players in the Authentication/Authorization Web service procedure.

**Step 1: User login.** The user has a requirement to access a data resource, so she uses a SOAP client to login with a username and password pair. The client passes the pair (and potentially other security factors) to the LDAP for authentication. LDAP validates the user and returns the information to the client.

**Step 2: SAML token.** The SOAP client passes the returned LDAP information and other user information to the SAML client, which is a service that packages the information in the correct format as a SAML token. The SAML client returns the token to the SOAP client. The SOAP client wraps the SAML token and the user request into a SOAP request.

**Step 3: SAML session begins.** The SOAP client starts a timed SAML session and sends the SOAP request to the appropriate Web service to fulfill the user requirement. The Web service parses the SAML token and verifies the authentication information via LDAP. In this way, the user only logs in once: Each Web service can serve as a user surrogate and is trusted by the data resource because the Web service verifies request information.

**Step 4: Authorization.** After authentication, the Web service sends a request to the SAML server that's running the rules engine. SAML isn't absolutely needed at this step, but it was convenient for us. The rules engine evaluates user parameters and determines the level of access that the user is authorized for. The evaluation is based on a set of rules that reflect predefined access policies. Verification of access level is returned to the Web service.

**Step 5: Request fulfillment.** The Web service requests data from the data resource, packages the result as a SOAP response, and sends it back to the SOAP client. The SOAP client presents the data to the user and terminates the SAML session. The SAML token expires and can't be reused.

## Centralized = Unified

To summarize, we built a SAML-based AA Web service that depends on LDAP as a centralized authority. Use of LDAP and an authorization rules engine that enforces security policies achieves centralized identity management. One benefit of centralized identity management is simplicity and unified access-control management.

Centralizing identity management means that you can manage access-control lists at one point (for example, termination of an employee account results in termination of all access accounts associated with the central account, in this case, LDAP). This Web service provides a generalized service for assertions, certification, and access. Developers can call this Web service to allow their Web services to gain access to targeted applications and data resources.

## Lessons Learned

SAML transports a token. It isn't the vehicle for authentication or authorization. To mitigate risk, SAML systems use timed sessions, IP addresses, HTTPS, and SSL/TLS. A security Web service is a very complex project with legal, security, and political facets.

Use of LDAP as a centralized authority achieved centralized identity management. The rules engine provides a powerful, flexible approach to access-control and policy management. (It supports both role-based and attribute-based access control.)

Identity management is simplified, first, because accounts can be activated and deactivated at the LDAP (for initiation and termination). Second, you avoid costs by centralizing and reducing authentication applications, having to increase administrative headcount to produce a positive ROI, and reducing the risk of audits. And finally, identity management is simpler because account management uses existing, established LDAP service, reducing overhead and increasing synchronization.

## Page 3

## Resources

**Cohen, F. "Debunking SAML Myths and Misunderstandings," IBM Developerworks Web site,SAML Myths**

[www-106.ibm.com/developerworks/xml/library/x-samlmyth.html?Open&ca=daw-se-news](http://www-106.ibm.com/developerworks/xml/library/x-samlmyth.html?Open&ca=daw-se-news)

**"Who Are You?" Sept. 1, 2003**

[www.intelligententerprise.com/030901/614feat3\\_3.jhtml?\\_requestid=602630](http://www.intelligententerprise.com/030901/614feat3_3.jhtml?_requestid=602630)

**Security Assertion Markup Language (SAML)**

[www.oasis-open.org/committees/download.php/6837/sstc-saml-tech-overview-1.1-cd.pdf](http://www.oasis-open.org/committees/download.php/6837/sstc-saml-tech-overview-1.1-cd.pdf)

**Electronic Authentication Partnership (EAP)**

[www.eapartnership.org](http://www.eapartnership.org)

**Jericho Systems**

[www.jerichosystems.com](http://www.jerichosystems.com)

**RSA Security**

[www.rsa.com](http://www.rsa.com)

### **Netegrity**

[www.netegrity.com](http://www.netegrity.com)

### **Oblix**

[www.oblix.com](http://www.oblix.com)

### **IBM**

[www.ibm.com](http://www.ibm.com)

### **Entrust**

[www.entrust.com](http://www.entrust.com)

### **OASIS, SAML information**

[lists.oasis-open.org/archives/wss/200403/bin00000.bin](http://lists.oasis-open.org/archives/wss/200403/bin00000.bin)

## **Vendor Trends in Identity Management**

Identity management is a complex issue. It includes password maintenance, provisioning management, and access-control management. This article focuses on a portion of identity management that is greatly facilitated by SAML, specifically role-based access control (RBAC). An increasing number of vendors support RBAC, which allows administrators to manage access-control lists more simply because users are grouped by categories or roles. Managing access-control lists for a few roles is simpler than managing lists for a few hundred users.

This article discusses an implementation with Jericho Systems' EnterSpace. Here's a short list of other prominent vendors:

### **Entegrity**

Entegrity offers the SAML-based product, AssureAccess. This Java-based access management software protects portals and Web services access. The product includes LDAP-based authentication, single sign-on, authorization, audit, user management, and security policy administration out of the box.

### **Entrust**

Entrust's Secure Identity Management Solution has modular access management and identity management components that can be mixed and replaced to support user requirements. Entrust's SAML-based GetAccess centralizes security management to provide a common infrastructure to manage user identities and enable authentication and authorization across multiple applications.

### **IBM**

IBM (along with ActivCard, Bioscrypt, ImageWare, and VeriSign) has announced a new Identity Management Portfolio designed to help organizations protect their information resources by incorporating additional layers of authentication and authorization into everyday business processes. The IBM collaboration simplifies identity management and reduces overall cost by offering a unified system that protects a wider range of identity management systems, from data, computers, and networks to employee badges, door locks, and security cameras.

### **Identrus**

The Identrus System standardizes digital identity authentication so that financial institutions can provide online services to their customers. The Identrus System enables global financial institutions to build

trusted interrelationships that offer third-party services to their business customers.

### Intrusic

Intrusic focuses on internal threats from hackers within the firewall. Its product, Zephon, identifies compromises to internal security by using a multilevel analysis architecture, which highlights inconsistencies in internal information flow. The assumption is that internal attacks create exception conditions in normal network operations and that these exceptions can be tracked and shut down. Zephon locates these exceptions by analyzing the environment, host, session, and packets of an information flow interaction.

### Netegrity

Netegrity's Siteminder offers access management tools with role-based access control. The company's TransactionMinder is a full identity management and access management package that offers out-of-the-box SAML-based policy management security for Web services. SAML is modeled after Netegrity's work in XML-based security for authentication and authorization, defined in the Security Services Markup Language specification. (Note: In October 2004, Computer Associates announced that it would acquire Netegrity.)

### Oblix

CORE and Netpoint systems from Oblix cover various facets of identity management. COREid supports identity management and policy management with integrated provisioning, access-control, and compliance-reporting packages. COREsv 4 is an enterprisewide Web services security and management deployment package, which includes COREid. The company's NetPoint is a SAML implementation for single sign-on that has been used by Southwest Airlines and the U.S. Navy.

### RSA Security

RSA offers a variety of products related to identity management. RSA's ClearTrust is a rules-based platform that provides the capability for Web access management, supported by a centralized policy management function. The use of SAML in ClearTrust provides better identity management, authentication, and single sign-on across organizations. RSA is a strong supporter of SAML. It recently granted royalty-free access to two key patents involved in SAML technology.

### Sun Microsystems

Sun ONE Identity Server uses SAML to support an out-of-the-box specification called "Liberty Alliance." This provides identity management and includes access management, identity administration, and enforcement of authentication and access policies. — *Hank Simon*

*Hank Simon, a member of W3C and OASIS, leads the Web Services Technical Advisory Group, which is a 100 person, cross-line business team of Web services/SOA strategists, architects, developers, and implementers. Simon has been designing and developing IT architectures for 27 years. He has published more than 100 articles and six books on XML, Web services, and advanced technologies.*

© 2006 CMP Media LLC

[Return to Article](#)