GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at http://www.giac.org/registration/gsec

# SAML: Common Security Language for Web Services

**Artyom Poghosyan**

**GSEC Version 1.4b option 1**

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

**1 Introduction.....3**

**2 Web Services Business Context.....4**

*2.1 Business Drivers.....4*

*2.2 Roadblocks to Web Services Enablement.....4*

**3 Security As Enabler of Web Services.....5**

*3.1 Security Objectives of Web Services.....5*

*3.2 Emerging Standards As Building Blocks for Solution Frameworks.....6*

3.2.1 W3C Web Services Standards.....6

3.2.2 OASIS Security Standards.....6

3.2.3 Other Key Web Services Standards.....7

**4 SAML Architecture.....8**

*4.1 How it Works.....8*

*4.2 SAML and WS-S: Friend or Foe.....10*

**5 SAML Security Considerations.....11**

*5.1 Denial of Service (DoS) Attacks.....11*

*5.2 Eavesdropping.....11*

*5.3 Replay Attacks.....12*

*5.4 Message Deletion / Modification.....12*

*5.5 Man-in-the-Middle Attacks.....12*

**6 SAML Implementations.....13**

**7 Conclusion.....14**

**References.....14**



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

# 1 Introduction

Web services represent the natural advancement of distributed computing to the next level on its evolutionary path. By definition, web services "*are self-contained, self-describing, modular applications that can be published, located, and invoked across the web"* [1, Vasudevan, 2001]. These heterogeneous applications relate to each other using XML (Extensible Markup Language) as a common language. Although the architectural frameworks of web services are presently in their embryonic stage, it is clear that two concepts will become key to their advancement: *interoperability* and *trust*.

The concept of interoperability acquires a new expanded meaning in the web services environment as the need grows for language-neutral, platform-independent applications capable of crossing organizational borders to conduct transactions across the web. Historically, a number of standards have been developed to facilitate the functionality and performance aspects of web applications. In fact, this is not an area where web services advancement faces major roadblocks.

What may become a serious inhibitor to the realization of web services potential is the lack of adequate trust mechanisms. In the context of web services this means open standards that:

a. serve the fundamental objectives of authentication, authorization, confidentiality, integrity and availability;
b. are accepted and widely deployed by players in the web services space.

It is this area that is significantly behind due to the lack of collaboration in the past towards development of common standards and protocols. What was mostly collaboration among groups of business partners and allies is becoming a cross-industry, cross-organizational effort to develop common standards that will enable trust relationships in web services. Some examples of collaboration include the Liberty Alliance project, OASIS, W3C, IETF, and Microsoft's Passport.

Although the effort is still on going, some critical milestones have already been achieved in establishing the basic building blocks upon which the framework of trust can be built. The Security Assertions Markup Language (SAML, pronounced "sam-el") is one such achievement.


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

# 2 Web Services Business Context

## *2.1 Business Drivers*

The constant search for new and creative ways to increase revenues, reduce costs and achieve business goals made the Internet revolution a reality. There is little doubt that the benefits derived have been sizable, both for business and consumers. However, many people would agree that the "e-game" is not over as there are still plenty of untapped opportunities the web has to offer. According to visionaries of the Internet and World Wide Web the next wave of the Internet revolution is approaching.

The emerging web services mechanism is likely to drive the next "e-revolution" due to its inherent features. Web services will provide what businesses and consumers require for more cost effective, process efficient and convenient online transactions:

- Web-based infrastructure for communication, service search, request and delivery
- Web single sign-on and federated identity management
- Seamless integration of proprietary-built business applications
- Integration of disparate computing resources into a single infrastructure

This list is not exhaustive and will continue to grow as new opportunities for utilization of web services are discovered. However, one of the most critical reasons for broad adoption of web services mechanism will be its trust-based architecture: one that will ensure secure and reliable transfer and storage of information assets. Therefore, it is of critical importance to have a coherent and robust security standards in the foundations of web services architecture.

## *2.2 Roadblocks to Web Services Enablement*

The process of developing web services technologies has had a rapid progress due to a number of mature technologies that currently support the web environment. However, independent research, such as that conducted by ZapThink, reports that security is the number one inhibitor of web services adoption at present [2, Bloomberg, 2002].

To some extent, web services security challenges are similar to those presented by existing web-based applications. On a high level, secure web services architecture must respond to the following concerns:

- Provide security solutions that can scale to increasingly open web services architecture and protect against new vulnerabilities. These solutions need to provide not only transport level security for point-to-point connections but also end-to-end application level security for multi-hop connections.


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- Ensure the digital integrity of transactions and confidentiality of proprietary information being transferred across the web.
- Address privacy concerns of consumers and allow regulatory compliance.

These are only a few key requirements among others that web services are expected to deliver. The next section will expand on critical security objectives as they relate to the web services environment.

# 3 Security As Enabler of Web Services

## *3.1 Security Objectives of Web Services*

Security becomes the platform upon which trusted web services can be deployed. In order to enable such services the fundamental security concepts must be built into the architecture of web services.

***Authentication*:** Web services must be able to positively identify the services they are communicating with. Authentication may be in one direction or bilateral and may take place in human-to-machine as well as machine-to-machine contexts. Web services must also support various authentication mechanisms.
***Authorization and access controls:*** These two objectives are critical because of new levels of access that are capable with web services. In addition to authorizing what information users/applications have access to, there also needs to be authorization of which operations an application or user has access rights to perform. Web services are programmatic interfaces and thus can be harder to monitor for suspicious activity. Administration of access controls will be an increasing challenge because web service environments are dynamic, heterogeneous in implementation and decentralized in architecture and administration. Since web services enable much easier integration with 3rd parties including suppliers, customers and partners [that may also be competitors], access rights must be tightly controlled and kept up-to-date.
***Confidentiality and integrity:*** Data exchange must be protected from modification while stored or in transit. Although encryption-based technologies like PKI and digital certificates are capable of safeguarding the data, web services present a new challenge. For example, it is possible that digital signatures or public keys need to be applied by multiple parties to different information components within a service.
***Non-repudiation:*** Web services must provide mechanism of message source verification to allow for non-repudiation.
***Privacy:*** Communications among web services must be safe from eavesdropping and unauthorized disclosure of data.


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

### *3.2 Emerging Standards As Building Blocks for Solution Frameworks*

This section briefly introduces the key web services standards and the collaborative organizations that drive the process of defining standards for web services security.

#### 3.2.1 W3C Web Services Standards

World Wide Web Consortium (W3C) is a leading developer of interoperable standards for the web. Simple Object Access Protocol (SOAP) is one of the key interoperability standards for web services developed by W3C.

c **SOAP** is an XML based protocol that connects web services to each other using HTTP as a transport. It defines mechanism for performing remote procedure calls (RPCs) and transferring EDI-style documents from one SOAP-enabled application to another.

W3C has also developed three main XML-based security components that are becoming basic building blocks for secure web services architecture:

c **XML Signatures (XML DSig)** defines a standard for achieving authentication and ensuring integrity using digital signature concepts applied to XML applications.
c **XML Encryption (XML Enc)** is a standard that allows for XML documents to be encrypted and thus protects the confidentiality of the information being transferred.
c **XML Key Management Specification (XKMS)** defines standard method for encryption key pair generation, key registration, public key validation and public key revocation. This way XKMS bridges the gaps that exist in prior two specifications as well as simplifies the PKI operations of XML-based clients.

Each of these specifications relies on traditional PKI foundations and concepts, but introduces two important changes: (1) authenticating and validating credentials of both the client and the server (which is uncommon in today's web applications) and (2) offloading the heavy processing of client-side PKI functions to services in a server environment – allowing thin XML-based clients to complete complicated trust services [3, RSA Security, 2003].

#### 3.2.2 OASIS Security Standards

The Organization for Advancement of Structured Information Standards (OASIS) is a non-profit, international consortium that creates interoperable industry specifications based on public standards such as XML. Two main web services security standards are being developed by OASIS: Security Assertion Markup Language (SAML) and Web Services Security (WS-S). These standards


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

propose a framework for conducting a trusted transaction, and can incorporate XML security components developed by W3C.

c **SAML** v1.0 was ratified by OASIS as an industry standard in November 2002. SAML defines a service that enables SOAP messages to pass "assertions" between two parties: an "assertion authority" and a "relying party" (see Figure 1). Security information is transmitted from one application to another as a vendor-independent XML document. The SAML specification does not define new technology for authentication or authorization, but provides a common language using XML to describe the information generated by systems across the Internet. A more detailed discussion of SAML architecture is provided in section 4 of this paper.

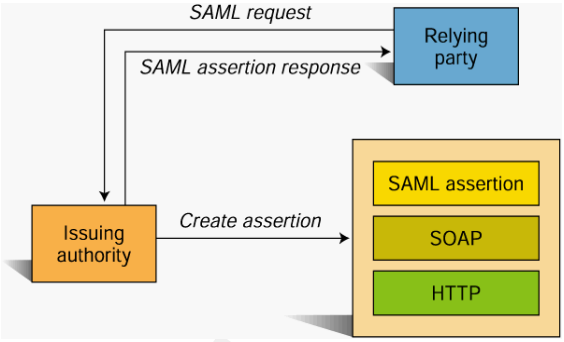

Figure 1. SAML Architecture (Source: Johnston,2002)

c **WS-Security (WS-S)**. IBM, Microsoft and VeriSign proposed WS-Security specification in April 2002 [4, Della-Libera et al, 2002] [5, Atkinson et al, 2002]. In June 2002 the three companies submitted the specification to OASIS for consideration as a standard. WS-Security is an XML- and SOAP-based message security model that provides broad set of specifications covering authentication, authorization, privacy, trust, integrity, confidentiality, secure communication channels, identity federation, delegation, and auditing of web services.

WS-S consists of three main components: how to define authentication, support digital integrity and protect message confidentiality:

- WS-S does not define what authentication mechanism must be used, but it does define its own language to complete authorization/authentication called Extensible Rights Markup Language (XrML).
- WS-S ensures the integrity of all or part of a message based on XML Signature. It can support multiple digital signatures.
- WS-S uses XML Encryption to protect the confidentiality of all or part of a message.

Although WS-S and SAML share similar concepts, the two are viewed as complementary rather than competing standards. Comparison of the two specifications in section 4 will help identify key differences between SAML and WS-S.

### 3.2.3 Other Key Web Services Standards


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

c Web Service Description Language (**WSDL**) is a protocol used to define how a service subscriber will interface with the defined service. WSDL uses XML and SOAP to describe the web service interface in the form of an XML document referred to as a WSDL file.

c Universal Description Discovery and Integration (**UDDI**) provides a method for publishing service descriptions in order to allow for web services to be located and accessed.

# 4 SAML Architecture

## *4.1 How it Works*

At present, web-based single sign-on is performed using transient browser session cookies. Interoperability becomes a problem in the current model because each security solution uses a different cookie security mechanism and encryption algorithm. SAML provides an interoperable alternative mechanism. It enables passing credentials and other related information in the form of "assertions" between sites, that have individual authentication and authorization systems, in order to create a single sign-on trusted environment among heterogeneous applications (see Figure 1 on p. 7).

```
<samlp: Request ...>
 <samlp: AttributeQuery>
    <saml: Subject>
     <saml: NameIdentifier
       SecurityDomain="sun.com"
       Name="rimap"/>
    </ saml: Subject>
  <saml: AttributeDesignator
      AttributeName="Employee_ ID"
      AttributeNamespace="sun.com">
  </ saml: AttributeDesignator>
 </ samlp: AttributeQuery>
</ samlp: Request>
```

Figure 2. Authentication Request (Source: Patel Sriganesh, 2002)

According to Rima Patel, "an assertion is a declaration of a 'certain fact' about a subject (user or code) that an individual was authenticated by a particular method at a specific time, or that an application has been granted a certain class of access to a resource under certain conditions" [6, Patel Sriganesh, 2002]. A SAML-compliant service, called a *Relying Party*, sends SAML *Requests* (Figure 2) to an *Issuing Authority*, which returns SAML *Assertion Responses* (Figure 3). SAML embeds the request-response messages in a SOAP envelope, which in turn, is transmitted via HTTP. The current SAML specification defines three main types of assertions:

- Authentication

```
<samlp: Response
  MajorVersion="1" MinorVersion="0"
  RequestID="128.14.234.20.90123456"
  InResponseTo="123.45.678.90.12345678"
  StatusCode="Success">

  <saml: Assertion
  MajorVersion="1" MinorVersion="0"
  AssertionID="123.45.678.90.12345678"
  Issuer="Sun Microsystems, Inc."
  IssueInstant="2002- 01- 14T10: 00:23Z">
  <saml: Conditions
  NotBefore="2002- 01- 14T10: 00: 30Z"
  NotAfter="2002- 01- 14T10: 15: 00Z" />
  <saml: AuthenticationStatement
  AuthenticationMethod="Password"
  AuthenticationInstant="2001- 01- 14T10:
  00: 20Z">
  <saml: Subject>
    <saml: NameIdentifier
      SecurityDomain="sun. com"
      Name="rimap" />
  </ saml: Subject>
  </ saml: AuthenticationStatement>
  </ saml: Assertion>
</ samlp: Response>
```

Figure 3. Authentication Response (Source: Patel Sriganesh, 2002)

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- Attribute
- Authorization decision.

It is important to note that SAML assertions do not actually perform authentication. Instead, they encapsulate the authentication process and provide transport for it. The SAML1.0 specification provides for multiple authentication-issuing authorities, which means that a service can determine what authentication mechanism to use, whether it is PKI, Kerberos, password-based or something else.

A typical SAML assertion consists of several common elements:

- Issuer ID and issuance timestamp
- Assertion ID
- Subject
  - ✓ Name and security domain
  - ✓ Subject's authentication data
- Advice (optional information provided by the issuing authority)
- Conditions under which the assertion is valid (NotBefore, NotOnOrAfter)
- Audience restrictions
- Target restrictions (intended URLs for the assertion)
- Application specific conditions.

SAML Use Case Scenario

The following use case scenario for employing SAML in a web services system explains how SAML assertions are produced and consumed. Vendor A makes a service available on the web that vendor B decides to incorporate into its own web offering. Users who log in to vendorB.com site might be using the web service from vendor A without being aware of it. Preferably, the entire process would be transparent to the user. However, if vendor A's service requires separate authentication, that transparency would be weakened, which could bother users. A system where the user's login to vendorB.com were passed along to vendorA.com would be far better. Vendor A would agree to trust users automatically whom vendor B already authenticates. Here is how the process would flow in the later case [7, Netegrity, 2001]:

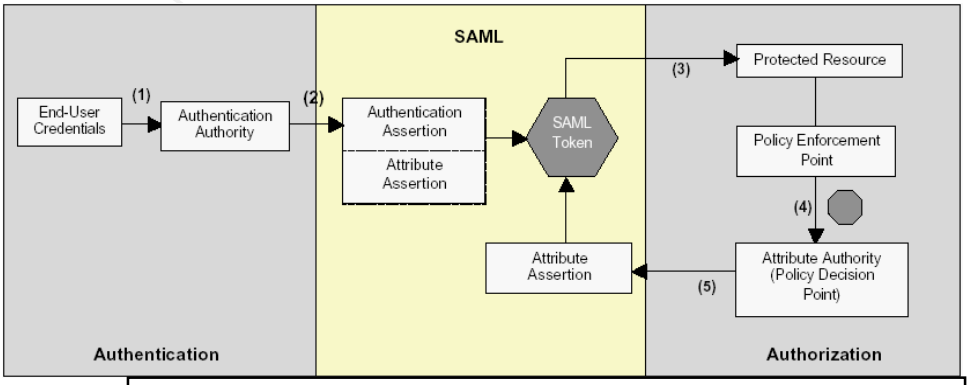


Figure 4. SAML Assertions Producer and Consumer Model (Source: Netegrity)



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

1. End-user submits login credentials to a SAML-enabled Authentication Authority application
2. Authentication Authority asserts user's credentials against user directory and generates an Authentication Assertion together with one or more Attribute Assertions (e.g., role and other user profile information). End-user is authenticated and identified by SAML assertions, which are assembled in a token.
3. End-user attempts to access a protected resource using her SAML token.
4. Policy Enforcement Point (PEP) intercepts end-user request to protected resource and submits the end-user's SAML token to the SAML-enabled Attribute Authority (security engine or business application).
5. Attribute Authority or Policy Decision Point (PDP) makes a decision based on its policies. If it authorizes access to resource, it then generates an Attribute Assertion attached to the user's SAML token. The end-user's SAML token can be presented to trusted business partners affiliated in a single sign-on relationship.

### *4.2 SAML and WS-S: Friend or Foe*

As mentioned above, WS-Security suggests a broad set of specifications that define security framework for web services. The WS-Security roadmap document [4, Della-Libera et al, 2002] describes six subsidiary specifications covering the details of security policy and federation (see Figure 5). None of these specs is yet a formal proposal, but it is expected that detailed specifications will be submitted to a standards body for consideration.

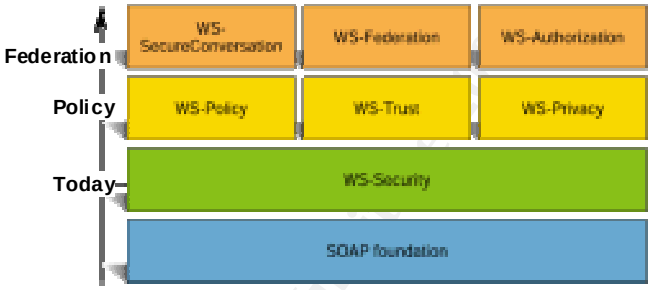

Figure 5. WS-S Specifications (Source: IBM/Microsoft)

Clearly, SAML and WS-S share common goals. However, a major goal of WS-S is "to be neutral with respect to identity assertion mechanisms and protocols" [4,Della-Libera et al, 2002].

According to draft specifications, WS-Security will function at a higher level of abstraction, enabling SAML assertions to be included as a supported technology for expressing security credentials along with other technologies (e.g. Kerberos tickets, digital signatures, PKI or simple login ID-password combinations).

Like SAML, WS-Security uses SOAP messages to convey claims about security. Hence, SAML assertions can be embedded in SOAP messages of WS-Security framework. The XML header block `<wsse:Security>` defined in WS-S draft specification [8, Hallam-Baker et al, 2002] will be used for that purpose. Nonetheless, it is expected that not all WS-Security implementations will take advantage of


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

SAML and vice versa. The developer preference and interoperability will remain key. It is expected that OASIS will play a role in standardizing where SAML and WS-S overlap, but perhaps focusing on how applications can interoperate regardless of their use of SAML or WS-S.

# 5 SAML Security Considerations

Although SAML is intended to assist the establishment of trusted communication there are several security risks related to implementation and use of the standard. The following discussion will highlight some critical security threats that either the SAML protocol itself or the protocol bindings are susceptible to. For this purpose, it is important to note that protocol's current specification provides only for SOAP bindings for implementation of the SAML request-response protocol.

## *5.1 Denial of Service (DoS) Attacks*

According to the draft Security and Privacy Considerations for SAML document [9, Moses et al., 2002] SAML is vulnerable to DoS attacks. Due to the processing cost required for SAML requests (including parsing of request and assertion construction), the responder can potentially be flooded with requests, which do not take nearly as much effort to construct. The above-mentioned document suggests a combination of methods to counter this vulnerability:

- Requiring clients to authenticate at some level below the SAML protocol level (for example, using the SOAP over HTTP binding, with HTTP over TLS/SSL, and with a requirement for client-side certificates that have a trusted Certificate Authority at their root)
- Requiring requester to sign the request. This should lessen the order of the asymmetry between the work done by requester and responder thereby decreasing the risk of DoS attack.
- Limiting ability to issue SAML service requests to a set of known parties. This reduces the risk of a DoS attack since only attacks originating from within the finite set of known parties are possible.

The following group of vulnerabilities is relevant to the SAML SOAP protocol bindings. Since SAML SOAP binding does not require authentication and has no requirement for in-transit message confidentiality and integrity, it is open to a number of common attacks.

## *5.2 Eavesdropping*

It is possible that an eavesdropping party could acquire both the SOAP message containing a request and the SOAP message containing the corresponding


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

response. This acquisition exposes both the nature of the request and the details of the response, possibly including one or more assertions. A possible countermeasure is to provide some type of in-transit confidentiality. At the SOAP level this would mean constructing the message such that no one other than the intended party could access the message contents. XML Encryption is likely to be the solution for this problem. However, until XML Encryption is widely supported, HTTP over SSL/TLS can be used as one method.

### *5.3 Replay Attacks*

At the SOAP binding level the primary concern about replay is the potential for use of replay as a denial-of-service attack method. Note that XML Encryption does not help prevent from capturing and reuse of the message for replay. If an attacker captures a SAML request that has been signed and encrypted he can replay that request at any time without needing to be able to undo the encryption. This is a particular issue since the SAML request does not include information about the issue time of the request, thus making it difficult to determine if replay is occurring. One possible solution is to design systems that use the unique key of the request (its `RequestID`) to determine if this is a replay request or not.

### *5.4 Message Deletion / Modification*

A message deletion attack would prevent either a request or a response from reaching an addressee. SOAP binding does not address this threat, however, implementation of reliable messaging extensions will help reduce the risk.

Modification of the request to alter the details of the request can result in significantly different results being returned, which in turn can be used by a clever attacker to compromise systems depending on the assertions returned. For example, altering the list of requested attributes in the `<AttributeDesignator>` elements could produce results leading to compromise or rejection of the request by the responder. Similarly, modification of the assertion details can result in serious compromise. For example, altering authentication or authorization information in the assertions may result in serious security breaches. These potential threats can be addressed using an in-transit message integrity solution. At the SOAP binding level, use of the XML Signature for digitally signing requests and responses can help accomplish this goal.

### *5.5 Man-in-the-Middle Attacks*

The SOAP binding is vulnerable to this threat. It can be addressed using a bilateral authentication system between the parties, which would ensure that what the parties are receiving during the conversation comes from a trusted party. Although this method does not prevent from eavesdropping, the communication content is prevented from being altered.



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

# 6 SAML Implementations

The first public demonstration of SAML v1.0 -compliant products took place at the SAML Interoperability Event in July 2002 sponsored by Burton Group. Twelve vendors, including IBM, Novell, Oblix, Sun Microsystems, Baltimore Technologies, CrossLogix, Entegrity Solutions, ePeople, Overxeer, Netegrity, RSA Security, and Sigaba demonstrated interoperability of their products with the standard. However, only a handful of vendors have officially released SAML-compliant products. The following is a list of some major products that are known to the author at the time of writing this paper.

**Sun ONE Identity Server 6.0** (from Sun Microsystems) is a standards-based product designed to help organizations manage secure access to web and non web applications both on the Internet and extranets.

**GetAccess 7.0** (from Entrust) is a comprehensive web access management solution with plug-in authentication, authorization and administration services.

**SiteMinder 5.5** (from Netegrity) provides platform for single sign-on, authentication management, and entitlement management. It enables a company to create a SAML-based identity and share that SAML identity with a partner e-business site.
**Affiliate Agent** (from Netegrity) enables partner sites to more easily recognize and authenticate the SAML-based identities.

**NetPoint 6.1** (from Oblix) provides single sign-on across multiple Web-based applications, controls access to appropriate web-based applications and content, based on centralized security policies and user identity profiles.

**ClearTrust** (from RSA Security) is a web access control solution that centrally controls and manages user access privileges to web-based resources based on definable user attributes, business rules and security policies.

**AssureAccess** (from Entegrity) is an access management software that protects Java/J2EE-based web portals, and web services. It allows application developers deploying secure e-business solutions to include authentication, single sign-on, authorization, audit, user management and security policy administration in their applications without building custom security code for each application.

**SelectAccess Version 5.0** (from Baltimore Technologies) provides web-based single sign-on for a seamless user experience. SelectAccess helps reduce administration cost and complexity by providing a unified approach to defining authorization policies and securely managing role-based access to on-line resources.


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

# 7 Conclusion

Standards will remain a "moving" target as web services platforms continue to evolve. Many solution vendors have already committed resources to implementation of emerging open standards and it is expected that more products will be rolled out within the next few years. There is little doubt that advancement of the open security standards acts as a catalyst in this process and the SAML standard is one example.

Although the friction is inevitable among some major players with respect to the ownership of parts in the web services architectural framework, it is the author's opinion that the maturity of web services will only be achieved through collaborative effort around the underlying standards.

# References


1. Vasudevan, Venu. "A Web Services Primer", April 4, 2001
   URL: http://www.xml.com/pub/a/ws/2001/04/04/webservices/index.html

2. Bloomberg, Jason, "Report: XML and Web Services Security", ZapThink Research,
   URL: http://www.zapthink.com/reports/ZTR-WS104.html (June 20, 2002)

3. RSA Security, "Developer Spotlight: Web Services Security. Part One: A Web Services Security Primer", winter 2003 issue.
   URL: http://www.rsasecurity.com/newsletter/developer/2003_winter/developer.html

4. Della-Libera, Giovanni, et al. "Security in a Web Services World: A Proposed Architecture and Roadmap", Microsoft, IBM, VeriSign, April 2002
   URL: http://www-106.ibm.com/developerworks/library/ws-secure

5. Atkinson, Bob, et al. "Web Services Security (WS-Security)",
   Microsoft and IBM, April 2002
   URL: http://www-106.ibm.com/developerworks/webservices/library/ws-secmap/

6. Patel Sriganesh, Rima, "Implementing Single Sign-On in Java™ Technology Web Services (TS-2370)", Sun Microsystems, Inc, JavaOne Conference, March 29, 2002
   URL: http://servlet.java.sun.com/javaone/sf2002/conf/speakers/11086-bio.en.jsp

7. Netegrity White Paper, "The standard XML framework for secure information exchange Security Assertions Markup Language (SAML)", May 20, 2001

8. Hallam-Baker, Phillip et al., "Web Services Security SAML Token Binding", OASIS Working Draft 0 5, December 16, 2002
   URL: http://www.oasis-open.org/committees/wss/documents/WSS-SAML-05.pdf

9. Moses, Tim et al., "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML)", OASIS Committee Specification 01, May 31, 2002
   URL: http://www.oasis-open.org/committees/security/docs/cs-sstc-sec-consider-01.pdf

10. Farrell, Stephen, et al. "Assertions and protocol for the Security Assertion Markup Language", OASIS, May 2002




© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

URL: http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf

11. Byous, Jon, "Single Sign-on Simplicity with SAML: An Overview of Single Sign-on Capabilities Based on the Security Assertion Markup Language", May 9, 2002
URL: http://java.sun.com/features/2002/05/single-signon.html

12. Sutor, Bob, "Perspective: The five biggest myths about Web services", IBM,
URL: http://news.com.com/2010-1071-971149.html (November 26, 2002)

13. Wagner, Ray, "Web Services Security Q&A", Gartner Group, December 20, 2002
URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2907342,00.html

14. Johnston, Stuart, "Positive Identification", XML & Web Services Magazine, October/November 2002 Issue
URL:
http://www.fawcette.com/xmlmag/2002_10/magazine/features/sjohnston/default_pf.asp

© SANS Institute 2003, Author retains full rights.



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Last Updated: April 24th, 2017

# Upcoming Training

Click Here to
{Get CERTIFIED!}

| | | | |
|---|---|---|---|
| SANS Riyadh 2017 | Riyadh, Saudi Arabia | May 06, 2017 - May 11, 2017 | Live Event |
| SANS Security West 2017 | San Diego, CA | May 09, 2017 - May 18, 2017 | Live Event |
| SANS Northern Virginia - Reston 2017 | Reston, VA | May 21, 2017 - May 26, 2017 | Live Event |
| SANS Melbourne 2017 | Melbourne, Australia | May 22, 2017 - May 27, 2017 | Live Event |
| SANS London May 2017 | London, United Kingdom | May 22, 2017 - May 27, 2017 | Live Event |
| SANS Stockholm 2017 | Stockholm, Sweden | May 29, 2017 - Jun 03, 2017 | Live Event |
| Security Operations Center Summit & Training | Washington, DC | Jun 05, 2017 - Jun 12, 2017 | Live Event |
| Community SANS Boise SEC401 | Boise, ID | Jun 05, 2017 - Jun 10, 2017 | Community SANS |
| SANS Houston 2017 | Houston, TX | Jun 05, 2017 - Jun 10, 2017 | Live Event |
| Community SANS Ottawa SEC401 | Ottawa, ON | Jun 05, 2017 - Jun 10, 2017 | Community SANS |
| SANS San Francisco Summer 2017 | San Francisco, CA | Jun 05, 2017 - Jun 10, 2017 | Live Event |
| SANS Secure Europe 2017 | Amsterdam, Netherlands | Jun 12, 2017 - Jun 20, 2017 | Live Event |
| SANS Charlotte 2017 | Charlotte, NC | Jun 12, 2017 - Jun 17, 2017 | Live Event |
| SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style | Denver, CO | Jun 12, 2017 - Jun 17, 2017 | vLive |
| SANS Rocky Mountain 2017 | Denver, CO | Jun 12, 2017 - Jun 17, 2017 | Live Event |
| Community SANS Portland SEC401 | Portland, OR | Jun 12, 2017 - Jun 17, 2017 | Community SANS |
| SANS Minneapolis 2017 | Minneapolis, MN | Jun 19, 2017 - Jun 24, 2017 | Live Event |
| Community SANS Philadelphia SEC401 | Philadelphia, PA | Jun 19, 2017 - Jun 24, 2017 | Community SANS |
| SANS Paris 2017 | Paris, France | Jun 26, 2017 - Jul 01, 2017 | Live Event |
| SANS Columbia, MD 2017 | Columbia, MD | Jun 26, 2017 - Jul 01, 2017 | Live Event |
| SANS Cyber Defence Canberra 2017 | Canberra, Australia | Jun 26, 2017 - Jul 08, 2017 | Live Event |
| SANS London July 2017 | London, United Kingdom | Jul 03, 2017 - Jul 08, 2017 | Live Event |
| Cyber Defence Japan 2017 | Tokyo, Japan | Jul 05, 2017 - Jul 15, 2017 | Live Event |
| Community SANS Phoenix SEC401 | Phoenix, AZ | Jul 10, 2017 - Jul 15, 2017 | Community SANS |
| SANS Cyber Defence Singapore 2017 | Singapore, Singapore | Jul 10, 2017 - Jul 15, 2017 | Live Event |
| SANS Munich Summer 2017 | Munich, Germany | Jul 10, 2017 - Jul 15, 2017 | Live Event |
| Community SANS Minneapolis SEC401 | Minneapolis, MN | Jul 10, 2017 - Jul 15, 2017 | Community SANS |
| SANS Los Angeles - Long Beach 2017 | Long Beach, CA | Jul 10, 2017 - Jul 15, 2017 | Live Event |
| Mentor Session - SEC401 | Macon, GA | Jul 12, 2017 - Aug 23, 2017 | Mentor |
| Mentor Session - SEC401 | Ventura, CA | Jul 12, 2017 - Sep 13, 2017 | Mentor |
| Community SANS Colorado Springs SEC401 | Colorado Springs, CO | Jul 17, 2017 - Jul 22, 2017 | Community SANS |