WOSIS 2004
2nd Workshop on the Security of Information Systems

Towards a Systematic Development
of Secure Systems

Efficient Tree Search in
Encrypted Data

Web Services Security
Is the Problem Solved?

The Impact of Virus Attack
Announcements on the Market Value
of Firms

Fair Trading Protocol with Offline
Anonymous Credit Card Payment

Filtering Spam at E-Mail Server Level
with Improved CRM114

AUERBACH

(ISC)²
PRESS

Web Services Security: Is the Problem Solved?

Carlos Gutiérrez, Eduardo Fernández-Medina, and Mario Piattini

This paper demonstrates that much work needs to be done in Web services security standardization. It explains the new Web services security threats and mentions the main initiatives and their respective specifications that try to prevent them.

Recently Web services technology has reached such a level of maturity that it has evolved from being a promising technology to becoming a reality on which IT departments are basing their operations to achieve a direct alignment with the business operations that they support.[1] In fact, based on the most recent reports from IDC (International Data Group),[2] approximately 3300 Web services-based technology projects were deployed all over North America in 2002 and it is expected that the expenses will approach $3 billion in 2003. This seems to be a logical consequence of the numerous advantages offered by the Web services paradigm:

☐ Standard-based middleware technology,
☐ Business services high reusability level,
☐ Easy business legacy systems leverage, and
☐ Integration between heterogeneous systems.

Due to these immediate benefits, most IT departments are implementing this technology with the high-priority objective of making them operable leaving aside, at least until later stages, the problems related to security. In general, the industry is still reticent to incorporate this technology due to the inadequate understanding that they have of the security risks involved, and the false belief that they will have to make a costly reinvestment in their security infrastructures.

Web services as distributed decentralized systems that provide well-defined services to certain (or not) predetermined clients, must be concerned with typical security problems common to the communication paradigm, through a compromised channel, between two or more parties.

MAIN WEB SERVICES SECURITY ISSUES

The following section describes some of the major security issues that Web services technologies must address.

Authentication: Any Web service that participates in an interaction may be required to provide authentication credentials by the other party. When certain service A makes a request for a service to service B, the latter may require credentials along with a demonstration of

CARLOS GUTIÉRREZ is with Sistemas Técnicos de Loterías del Estado in Madrid, Spain. He may be reached at carlos.gutierrez@stl.es.

EDUARDO FERNÁNDEZ-MEDINA and MARIO PIATTINI are with the Alarcos Research Group at the Universidad de Castilla-La Mancha in Ciudad Real, Spain. They may be reached at Eduardo.FdezMedina@uclm.es and Mario.Piattini@uclm.es, respectively.

its ownership (e.g., a pair username/ password or an X.509v3 certificate).

Authorization: Web services should include mechanisms that allow them to control access to the services being offered. They should be able to determine who can do what and how on their resources.

Confidentiality: Keeping the information exchanged among Web services nodes secret is another of the main properties that should be guaranteed in order to consider the channel secure. Confidentiality is achieved thanks to ciphering techniques.

Integrity: This property guarantees that the information received by a Web service remains the same as the information that was sent from the client. A simple checksum might offer integrity when accidental changes are made in the data.

Nonrepudiation: In the Web services world, it is necessary to be able to prove that a client utilized a service (requester non-repudiation) and that the service processed the client request (provider nonrepudiation). This security issue is covered by means of digital signatures.

Availability: The need to take care of the availability aspects for preventing denial-of-service attacks or to arrange redundancy systems is a crucial point in Web services technology, above all, in those scenarios in which the services provide critical services: real-time services, Certification Revocation Lists services, and so on.

End-to-end security: Network topologies require end-to-end security to be maintained all across the intermediaries in the message's path. "When data is received and forwarded on by an intermediary beyond the transport layer, both the integrity of data and any security information that flows with it may be lost. This forces any upstream message processors to rely on the security evaluations made by previous intermediaries and to completely trust their handling of the content of messages."[3]

Up to this point, we have briefly reviewed the typical security problems tightly related to distributed computing systems. Web services must address both these, inherited from the distributed computing classical scheme, and, in addition, those arising from the new threats created by its own nature. Some of these new threats are:

☐ Diversity and a very high number of standard specifications that do not facilitate a clear vision of the problematic and its solutions;

☐ The current draft state in which the majority of the security specifications are found;

☐ The Internet publication of a complete and well-documented interface to back-office data and the company's business logic;

☐ New XML standard formats needed to structure the security data;

☐ Application-level, end-to-end, and just-one-context-security communications;

☐ Interoperability of the requirements and online security elements;

☐ Audit and automatic and intelligent contingency processes aimed at being machine-to-machine interactions not controlled by humans;

☐ A complex dependency network that can lead to the execution of a business process depending on an unknown Web services number; and

☐ Online availability management in critical business processes.

The remainder of this article is divided into five parts. In the first one, a brief review of the core specifications that support the technology at hand is given. Next, core security Web services specifications are explained, and unresolved issues not yet addressed by them are described. In the next parts, the main initiatives are introduced as well as the specifications related to the security in which they are involved. The last section shows how the numerous and, to a certain, extent uncontrolled specifications development and initiatives are already

WS-Security

<<specification>>
WS-SecureConversation
(from WS_SecureConversation)
latestStableRelease = Draft

<<specification>>
WS-Federation
(from WS_Federation)
latestStableRelease = Draft

<<specification>>
WS-Authorization
(from WS-Security)
latestStableRelease = Draft

<<specification>>
WS-Trust
(from WS-Security)
latestStableRelease = Draft

<<specification>>
WS-Policy
(from WS-Security)
latestStableRelease = Draft

<<specification>>
WS-Privacy
(from WS-Security)
latestStableRelease = Draft

<<specification>>
WS-PolicyAttachment
(from Policy)
latestStableRelease = Draft

<<specification>>
WS-SecurityPolicy
(from Policy)
latestStableRelease = Draft

<<specification>>
WS-PolicyAssertions
(from Policy)
latestStableRelease = Draft

<<specification>>
WS-Security
(from WS-Security)
latestStableRelease = 1.0

CoreSecurity

<<specification>>
XML Encryption
(from Specification)
latestStableRelease = 1.0

<<specification>>
XML Digital Signature
(from Specification)
latestStableRelease = 1.0

<<specification>>
XML Key Management System
(from XML Key Management System)
latestStableRelease = 2.0

XML-Signature XPath Filter
(from Specification)
latestStableRelease = 1.0

<<specification>>
XML Decryption Transform
(from Specification)

<<specification>>
W#C Canonical XML
(from Specification)
latestStableRelease = 1.1

OASIS

<<specification>>
XACML
(from Specification)
latestStableRelease = 1.0

<<specification>>
SAML
(from Specification)
latestStableRelease = 1.1

Liberty Alliance Project

Core

<<specification>>
SOAP
(from Specification)
latestStableRelease = 1.2

<<specification>>
WSDL
(from Specification)
latestStableRelease = 1.1

<<specification>>
XML
(from XML)
latestStableRelease = 1.1

<<specification>>
UDDI
(from Specification)
latestStableRelease = 3.0.1

causing collisions among solutions to similar security problems.

WEB SERVICES CORE STANDARDS

In this section, we take a look at the four fundamental standards involved in the creation of operational Web services. Figure 1 outlines the most important security specifications under development. They are grouped as:

☐ *Core:* Web services foundational specifications. These are the standards on which Web services are based.

- ☐ *Core Security:* Standards that provide the XML low-level security primitives such as ciphering and signing.
- ☐ *OASIS:* Security specifications developed by the OASIS standards body.
- ☐ *WS-Security:* Family of specifications developed by Microsoft and IBM which are under the OASIS standardization process.
- ☐ *Liberty Alliance Project:* Represents the group of specifications developed in the Liberty Alliance Project.

"Basic services, their descriptions, and basic operations (publication, discovery, selection, and binding) that produce or utilize such descriptions constitute the SOA foundation." Web services are built on an architecture SOA basis. In fact, Web services architecture is an SOA architecture instantiation.[4] For that reason, the fundamental characteristics described by SOA are the ones that have initially headed the major efforts in the industry standards development process. The core Web services specifications are XML,[5] SOAP,[6] WSDL,[7] and UDDI.[8]

These specifications have been broadly adopted by the industry, and constitute the basic building blocks on which Web services are being designed and implemented. The bad news is that these four operative services specifications allow the creation of Web services but they do not say anything about how to secure them. What's more, they themselves contain security questions that must be answered:

XML and SOAP: These specifications do not say anything about how to obtain integrity, confidentiality, and authenticity of the information that they respectively represent and transport.
UDDI and WSDL: Questions should be answered such as "Is the UDDI registry located in a trustworthy location? How can we be sure that the published data has not been maliciously manipulated? Was the data published by the business it is supposed to have been published by? Can we rely on the business that published

the services? Are the services available at any moment? Can we trust the transactions that are produced from the execution of the business services?" As we can see from all these questions, an in-depth analysis of the security problems that an UDDI and WSDL architecture implies is needed.[9] Despite all these drawbacks, these standards have evolved and matured and the industry has adopted and implemented most of them.

At this point, the main Web services standardization initiatives are the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). Both consortiums are trying to standardize their vision, security included, of what the Web services should be and should contribute to the WWW world. This parallelism is causing the existence of specifications developed by both groups that resolve similar problems.

As is expressed by IBM and Microsoft,[10] "We note that other organizations such as the IETF and ebXML are tackling a related set of problems, and we are pleased there are already formal liaisons between the W3C XML Protocol Working group and its counterparts in both ebXML and IETF."

All the involved groups will have to make efforts to unify in the future with the purpose of integrating their visions and standards and thus define a common and global framework.

CORE WEB SERVICES SECURITY STANDARDS

The W3C consortium is responsible for the development of the following XML technology standards: XML Encryption, XML Digital Signature, and XML Key Management System.

XML Encryption

W3C XML Encryption[11] has been a proposed standard since 2002. It provides a model for encryption, decryption, and representation of

full XML documents, single XML elements (and all descendants) in an XML document, contents of an XML element (some or all of its children including all its descendants) in an XML document, and arbitrary binary content outside an XML document.

XML Encryption solves the problem of confidentiality of SOAP messages exchanged in Web services. It describes the structure and syntax of the XML elements that represent encrypted information and it provides rules for encrypting/decrypting an XML document (or parts of it).

The specification states that encrypted fragments of a document should be replaced by XML elements specifically defined in the recommendation. In order to recover the original information, a decryption process is also specified.

Web services use XML for delivering the necessary metainformation (SOAP headers) and the payload. As a result, XML Encryption can be used for encrypting/decrypting any fragment or logical part of an XML message. XML Encryption does not specify how to encrypt SOAP messages generated by Web services. This task is delegated to higher-level specifications that would define rules for using this primitive within the information exchange context. XML Encryption also describes how to encrypt already encrypted content (superencryption) and provides a mechanism for encrypting the keys used in the process. Looking back at the beginning of this section, where a list is given of the data types that can be encrypted, we may miss the possibility of encrypting the tree nodes without having to encrypt full subtrees. Basically, the solution would consist of extracting the wanted nodes from the original document, encrypting each of them, and putting them in an encrypted nodes pool. The recipient will get the modified document and the encrypted nodes pool, and will be able to decrypt the nodes, which it is allowed to see, and put them back in place inside the document.[12]

One of the implicit security problems associated with this specification is the explicit declaration of the fragments that have been encrypted. According to the specification, information is encrypted and replaced by XML elements containing the result and so, analysis information attacks could be easily carried out on the output document.

Recursivity is also addressed, but no solution is given. Encrypted key A may need encrypted key B, but B may also need A. XML Encryption recommends the use of *ds:* namespace for these elements, which is what XML Digital Signature elements belong to, instead of providing a different namespace, as with the WS-Security family.

XML Digital Signature

XML Digital Signature[13] has been a W3C recommendation since 2002, the fruit of joint work between W3C and the IETF. It defines how to digitally sign XML content and how to represent the resulting information according to an XML schema. Digital signatures grant information integrity and nonrepudiation. Thus, for example, an entity cannot deny the authorship of a digitally signed bank transfer made through a Web service.

According to the XML Digital Signature specification, a digital signature can be applied to any kind of digital content, including XML. It can be applied to the contents of one or more resources. Enveloped signatures and enveloping signatures exist. Both are applied over data contained within the same XML document that carries the digital signature. Detached signatures that sign digital content not contained within the same XML document also exist.

Signature creation and verification processes are defined by the specification. It is, like XML Encryption, technology independent, so additional mechanisms are needed to define how it will be applied to Web services message exchange.

Applications using this specification combined with encryption must deal with some security-related issues. The following rules are proposed:

☐ When the data are ciphered, any digest or signs on the data would have to be ciphered as well so that it is prepared to anticipate guessing plaintext attacks.

☐ Use XML Decryption Transform transformation during the digital signature verification process.[14]

XML Key Management System

XML Key Management System[15] is a specification that has been subject to the W3C standardization process that proposes an information format as well as the necessary protocols to convert a Public-Key Infrastructure (PKI) in a Web service so that it will be able to register public/private key pairs, locate public keys, validate keys, revoke keys, and recover keys.

This way, the entire PKI is extended to the XML environment, thus allowing delegation of trustworthy decisions to specialized systems. XKMS is presented as the solution for the creation of a trustworthy service that offers all PKI subordinate services, but without resolving the inherent issues of the infrastructure.

☐ How can a Certification Authority's (CA) public key be known with total certainty? Is the CA-ascertained identity useful?

☐ There are known issues with OIDs (Object Identifiers) for automatic processing and their explosive and continuing growth.

☐ Because the global public key infrastructure is lacking a single world-recognized certification authority, it is not clear how to equip the world in order to allow two systems (e.g., Web services) that know nothing of each other to establish a trustworthy relationship through a third party on the fly and with no previous offline communication.

WEB SERVICES SECURITY: STANDARDS AND SECURITY ISSUES ALREADY ADDRESSED

Now that we have reviewed the basic Web services security standards and their related security, we detail the emerging technology and specifications that are based on these standards.

First, we briefly touch on the WS-* specifications, whose principal developers are IBM and Microsoft. Secondly and thirdly, we discuss the SAML and XACML standards, which have already been delivered by the OASIS organization in their initial versions, and whose objective is how to present information and the security policy, respectively. Fourthly, we briefly comment on the Liberty Alliance project, which is lead by Sun Microsystems, and fifthly and lastly, we give a summary in matrix form showing all the specifications covered in this article, noting those that have been delivered and those that are still in draft form.

WS-Security Family Specifications

IBM and Microsoft, together with other major companies, have defined a Web services security model that guarantees end-to-end communication security.

These companies are jointly elaborating on a series of specifications that compose an architecture, termed by Microsoft as Global XML Web Services Architecture,[16] that will lead the development in the Web services industry so that different products can interoperate within a secured context. The center of these specifications is composed of WS-Addressing, WS-Coordination, WS-Inspection, WS-Policy, WS-Referral, WS-ReliableMessaging, WS-Routing, WS-AtomicTransaction, and WS-Security.

We focus our attention on the last specification: WS-Security,[17] which IBM, Microsoft, and VeriSign developed and submitted to OASIS which is responsible for its standardization process. WS-Security "describes enhancements to SOAP messaging to provide *quality of protection* through message integrity, message confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies." This is the specification on which some additional specifications (some with publicized versions) that cover

all aspects of security in Web services have based their definition. WS-Security is placed at the base of the security specification pile. Its purpose is to provide quality of protection to the integration, adding the following properties to communication and messages: message integrity, confidentiality, and simple authentication of a message. WS-Security allows the easy incorporation of many existing security models such as PKI and Kerberos.

Other specifications that directly relate to security issues such as WS-SecurityPolicy, WS-Trust, WS-Privacy, WS-SecureConversation, WS-Authorization, and WS-Federation are being developed based on WS-Security.

In the protocol stack and right on top of WS-Security, we find the WS-Policy specifications (with its security attached WS-SecurityPolicy specification), WS-Trust, and WS-Privacy.

WS-Trust is another specification deserving mention due to its similarity with XKMS. WS-Trust defines an XML schema as well as protocols that allow security tokens to be accessed, validated, and exchanged. However, this is not a new problem because the XKMS specification already addresses it when the underlying security infrastructure is PKI. Therefore, if we wish to extend a PKI as Web service, which of the two standards should we use?

Another noteworthy specification is WS-Policy and its related specifications WS-SecurityPolicy, WS-PolicyAssertions, and WS-PolicyAttachment. These specifications define an XML syntax for defining Web service policies (WS-Policy); a way to relate policies to XML elements, UDDI entries, or WSDL descriptors; a combination of policy assertions of a general nature (WS-Policy-Assertions); and a combination of policy assertions of a security nature (WS-SecurityPolicy).

SAML

Secure Assertion Mark-Up Language[18] is an "OASIS Open Standard" specification developed by OASIS and was delivered in its first version in 2002.

Basically, this specification defines an XML schema that allows trust assertions (authentication, authorization, or attribute) representation in XML and request/response protocols to perform XML authentication, authorization, and attribute assertion requests.

However, SAML has not yet resolved all the problems related to interoperable XML security-data transferences.[19] However, it shows significant progress. For instance, SAML does not solve how the authentication evidence itself is transferred. This issue has been addressed by WS-Security through its UsernameToken and BinarySecurityToken security tokens definition. In addition, SAML does not define the way to include SAML assertions within SOAP "wsse:Security" block headers (defined by WS-Security specification). In August 2002, WS-Security specification delivered the technical paper "The WS-Security Profile for XML-Based Tokens"[20] in order to solve this matter.

XACML

XACML[21] has been another OASIS specification since February 2003 and its main intention is to define an XML vocabulary for specifying the rules by which access control decisions can be enforced.

XACML defines these access control rules depending on the requester characteristics, communication protocol in use, and the authentication mechanism used. XACML is very similar, as far as the security problem it solves, to the policy rules model and language defined by the previously studied WS-Policy family of specifications. This coincidence is another example of the unification effort proof that an attempt will have to be made in the future to define a sole model and related language policy in the Web services world. XACML defines a service architecture that must be implemented by fully fledged policy architectures.

FIGURE 2 XACML Policy Services Architecture
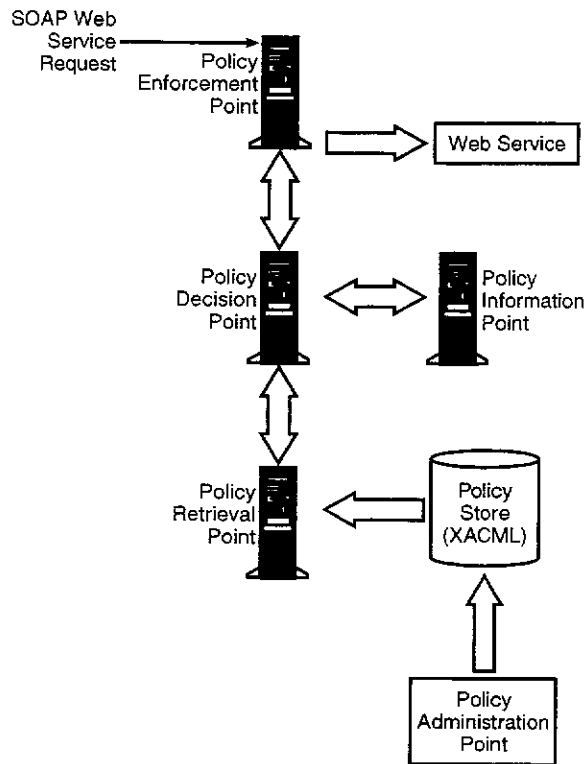
In Figure 2, services and conversations that take part in certain SOAP request authorization processes are shown.

The SOAP request that travels towards the Web service is intercepted by the PEP (Policy Enforcement Point) service whose task is to enforce the authorization decision on the request. PEP asks PDP (Policy Decision Point) for an authorization decision to be obtained. It is responsible for the PDP service, evaluating the policies, and subsequently answering to the PEP service as to whether the access is permitted. In some cases the PDP will need to obtain the policy information from a specialized PRP (Policy Retrieval Point) service node. This service will provide the PDP with suitable policies. In addition, the PDP may need some extra attribute-like information (SAML attribute assertions) about the requester, its environment, or the subject. If so, it will request this data from a PIP (Policy Information Point)

service. Once all the necessary data is gathered, the PDP will proceed to evaluate it and give a positive or negative access answer to the PEP that, subsequently, will enforce it. Finally, if access is permitted, the SOAP request will reach its destination Web service.

Liberty Alliance Project
The Liberty Alliance Project[22] is led by Sun Microsystems, and its purpose is to define a standard federation framework that allows services such as Single Sign-On.

Thus, the intention is to define an authentication distributed system that allows intuitive and seamless business interactions. As we can see in Table 1, this purpose is the same as those of the WS-Federation specification and Passport's .NET technology. Once again, this is another example of the previously so-called overlap problem in Web services security solutions.

TABLE 1 Summary of the Current Web Services Standard Development Efforts Grouped by Topic

Authentication	WS-Security, WS-Trust (Draft), XKMS, SAML Profiles (Request/Response Protocol for Obtaining SAML Assertions), Liberty Alliance Project (SSO Using Extending SAML Framework), WS-Federation (SSO) (Draft)
Authorization	XACML (policy-base authorization), WS-authorization (draft)
Confidentiality	W3C XML Encryption, WS-Security
Integrity	W3C XML Digital Signature
Nonrepudiation	W3C XML Digital Signature, WS-Security
Security policies	WS-Policy + WS-SecurityPolicy (draft), XACML
Trust authority	WS-Trust (draft); W3C XKMS
Security contexts/key derivation	WS-SecureConversation (draft)
Delegation/proxy	WS-Trust (draft), Delegation has not yet been fully addressed
Privacy	WS-Privacy (draft)
Attribute mapping	No addressee
Reference security architecture	No addressee
Security methodology	No addressee

ISSUES TO BE SOLVED

In spite of the amount of specifications that we have reviewed in this article, and summarized in Figure 1, there are many unresolved security issues that will have to be addressed and standardized in the future.

1. A clear effort should exist by all entities involved in this technology to unify their criteria and solutions. The explosion of specifications and concepts is such that the learning curve may become unacceptable for the most of the IT projects. As demonstrated in this article, questions such as knowing whether the chosen solution is the best of all the possible ones or, if a solution has been chosen, it will be supported long-term by the major industry companies, are difficult to answer.

2. Another problem to be solved is attribute or role principal mapping among different systems. Coherent access control decisions will be difficult to make when the same name of attributes or roles in both interacting Web services are set. For instance, a certain set of attributes assigned to user A in system Y may have a completely different meaning in another system, B. System B should need to map the attributes provided by user A to its own attributes types in order to be able to make a coherent access decision. RBAC together with a global attribute mapping agreement may be the way to reach a successful solution.

3. Nowadays, a methodology that accomplishes and considers all the possible security issues and defines an organized development process that directs Web services deployment in all expected (and unexpected) scenarios does not exist. This methodology should produce a distributed security framework. This framework should address all the necessary security primitives (authentication, security policy statements, confidentiality, ...) and should be flexible enough to allow primitive implementation solution replacements without affecting the overall performance of the system. Thus, it should be able to define a framework into which specialized security modules may be plugged. For instance, it should allow us to replace a WS-Trust security module for an XKMS module in a transparent way for the client. As a first approach, and inspired by SUN JMX architecture, we would design this framework by means of a security specialized microkernel creation in such a way. This microkernel would have a central component with no specific functionality beyond that of acting as a socket into which security modules can be plugged. Every security module

would plug in the socket by means of a well-known interface and would give notice to the component about the security primitives it provides. Any client security request would be intercepted by the central component and then redirected to the correspondence security service. The response would be brokered by the central component as well.

CONCLUSION

In this article, we have reviewed the current Web services security specification and initiatives and we have shown that its diversity is provoking an unclear vision of the problem and its solutions. In addition, unaddressed security issues have been stated overall and for each specification. The lack of a global standardization initiative is causing overlapping solutions to similar problems being put forward. This fact will require an extra effort in the future not only for the specifications to unify and make themselves interoperable but for industry to adopt and implement them.

Therefore, solutions to topics such as security policies, delegation, interbusiness principal attributes mapping, and privacy are not yet addressed by delivered and stable standards.

ACKNOWLEDGMENT

References

1. Casati, F., Shan, E., Dayal, U., and Shan, M.-C., Business-Oriented Management of Web Services. *Communications of the ACM*, 46(10), October, 25–28 (2003).
2. National Institute of Standards and Technology. Role-Based Access Control — Draft 4 April 2003 (2003). See *http://csrc.nist.gov/rbac/rbac-std-ncits.pdf*.
3. IBM and Microsoft. Security in a Web Services World: A Proposed Architecture and Roadmap — technical whitepaper 7 April 2002. See *http://msdn.microsoft.com/ws-security/*.
4. WSAS. Web Services Architecture Specification — WC3 Working Draft 8 August 2003 (2003). See *http://www.w3.org/TR/2003/WD-ws-arch-20030808/*.
5. W3C Extensible Markup Language (XML) 1.1 — W3C Recommendation 04 February 2004 (2004). See *http://www.w3.org/TR/xml11*.
6. SOAP Version 1.2 Part 0: Primer. See *http://www.w3.org/TR/2003/REC-soap12-part0-20030624/*
7. WSDL. Web Services Description Language (WSDL) 1.1 — W3C Note 15 March 2001. See *http://www.w3.org/TR/wsdl*.
8. UDDI Version 3.0.1 — UDDI Spec Technical Committee Specification 14 October 2003. See *http://uddi.org/pubs/uddi-v3.0.1-20031014.htm*.
9. Adams, C. and Boeyen, S. UDDI and WSDL Extensions for Web Services: A Security Framework. In *Proceedings of the ACM Workshop on XML Security*, Fairfax, VA, (2002).
10. IBM and Microsoft. Web Services Framework (2001). See *http://www.w3.org/2001/03/WSWS-popa/paper51*.
11. XML Encryption Syntax and Processing — W3C Recommendation 10 December 2002 (2002). See *http://www.w3.org/TR/xmlenc-core/*.
12. Geuer-Pollmann, C. XML Pool Encryption. In *Proceedings of the Workshop on XML Security*. Fairfax, VA, ACM, New York (2002).
13. W3C XML Signature Syntax and Processing — W3C Recommendation 12 February 2002 (2002). See *http://www.w3.org/TR/xmldsig-core/*.
14. W3C Decryption Transform for XML Signature — W3C Recommendation 10 December 2002 (2002). See *http://www.w3.org/TR/2002/REC-xmlenc-decrypt-20021210*.
15. W3C XML Key Management Specification (XKMS) — W3C Note 30 March 2001 (2001). See *http://www.w3.org/TR/xkms/*.
16. Box, D. (2002) Understanding GXA (2002). See *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngxa/html/gloxmlws500.asp*.
17. Web Services Security (WS-Security) — Specification 5 April 2002 (2002). See *http://www-106.ibm.com/developerworks/webservices/library/ws-secure*.
18. SAML. Assertions and Protocol for the OASIS 2 Security Assertion Markup Language 3 (SAML) V1.1 (2003). See *http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf*.
19. Harman, B., Flinn, D.J., Beznosov, K., and Kawamoto, S., *Mastering Web Services Security*, Wiley, New York (2003).
20. WS-Security Profile for XML-based Tokens — Specification 28 August 2002 (2002). See http://www-106.ibm.com/developerworks/webservices/library/ws-sectoken.html.
21. O'Neill, M., Hallam-Baker, P., Cann, S.M., Shema, M., Simon, E., Watters, P.A., and White, A., *Web Services Security*. McGraw-Hill, New York (2003).
22. Liberty Alliance Project. Introduction to the Liberty Alliance Identity Architecture (2003). See *http://www.projectliberty.org/resources/whitepapers/LAP%20Identity%20Architecture%20Whitepaper%20Final.pdf*.
23. RBAC. Role-based Access Control — Draft 4 April 2003. See *http://csrc.nist.gov/rbac/rbac-std-ncits.pdf*.
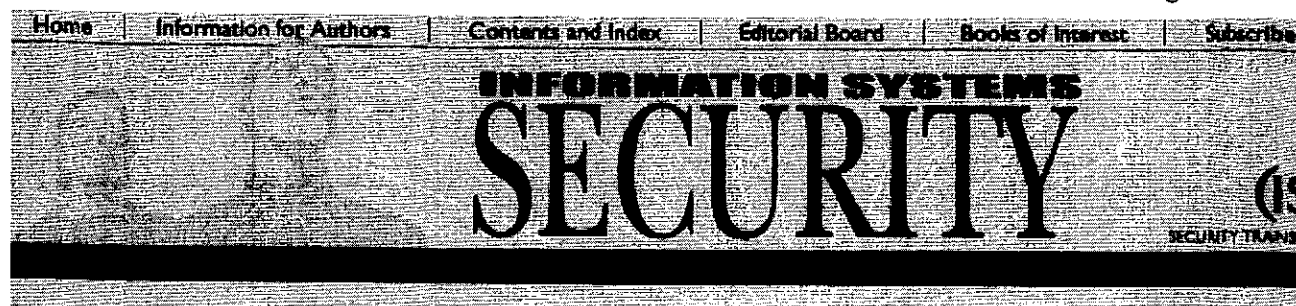
Information Systems Security Editorial Board

Managing Editor

Jim Tiller, CISM, CISA, CISSP, is Chief Security Officer and Managing Vice President of Security Services for International Network Services (INS). Jim has been with INS since 1998 and has provided security solutions for global organizations for the past 13 years. He is the author of The Ethical Hack: A Framework for Business Value Penetration Testing and A Technical Guide to IPSec Virtual Private Networks. He has published several papers on information security issues in the Information Security Management Handbook and in Information Systems Security.

Editor-at-Large

Rich O'Hanley

Editorial Advisors

Anton Chuvakin, Ph.D., GCIA, GCIH, is a senior security analyst with net- Forensics, where he is involved with designing the product and researching potential new security features. His areas of InfoSec expertise include intrusion detection, UNIX security, forensics, and honeypots, etc. He is the author of a book entitled Security Warrior and is a contributor to Know Your Enemy II and Information Security Management Handbook. In his spare time, he maintains his security portal at www.info-secure.org.

Todd Fitzgerald, CISSP, CISA, CISM, is the Director of Systems Security and Systems Security Officer for United Government Services, LLC, which is the largest processor of Medicare hospital claims on behalf of the Centers for Medicare and Medicaid Services (CMS) and is a subsidiary of WellPoint, Inc. He has over 25 years of broad-based information technology experience, holding senior IT management positions with Fortune 500 and Global Fortune 250 companies. Todd is a member of the board of directors and security taskforce co-chair for the HIPAA Collaborative of Wisconsin (HIPAA COW), a participant in the CMS/Gartner Security Best Practices Group, Blue Cross Blue Shield Association Information Security Advisory Group, previous board member for several Information Systems Security Associations (ISSA), and is a frequent speaker and writer on security issues. Todd focuses largely on issues related to security management, risk assessments, policy development, organizing security, security assessments, regulatory compliance (HIPAA, CAST, NIST, ISO 17799), security awareness, and developing security programs. He has published several papers on information security issues in the Information Security Management Handbook, in Information Systems Security, and in The HIPAA Program Reference Handbook.

Julio César Hernández Castro, Ph.D., is Associate Professor at Carlos III University in Madrid, Spain. He got a Mathematics degree, with specialization in Computer Science, in 1995. Then, he spent a number of years working as a security engineer and consultant for different companies, before obtaining a M.Sc. in Network Security and Coding Theory. After that, he joined Carlos III University where, since then, he has focused his research in the fields of cryptology, steganography & steganalysis, network & computer security (forensics, malware, attacks & defense, etc.) He has published more than 30 articles in International Conference and Journals, notably in Information Systems Security, Computational Intelligence, IEEE Computer, IEEE Latin America, Cryptologia, etc. He is the editor of various books, and has been guest editor and reviewer for a number of special issues of different journals, including the July/August 2004 issue of *Information Systems Security.* He is also involved with the organization of various International Conferences and Workshops.

Ralph Spencer Poore, CFE, CISA, CISSP, CHS-III, Principal Consultant, Innovè LLC and Senior Partner, Pi R Squared Consulting, LLP, provides security, privacy, and compliance consulting services, continuing a 30-plus year distinguished career in information security as an inventor, author, consultant, CISO, CTO, college instructor and entrepreneur. He has published widely including articles on information security issues in the Information Security Management Handbook and in Information Systems Security (where he was a past consulting editor). He served in numerous capacities with (ISC)² including as a past International President, as founding Chairman of the Test Development Committee, and as Chairman of the Governance Committee. He currently serves on the Professional Conduct Committee, the CBK Committee, and the Americas Advisory Board.

Ben Rothke, CISSP, CISM, is a New York City-based senior security consultant with ThruPoint, Inc., and has over 15 years of industry experience in the area of information systems security. His areas of expertise are in PKI, HIPAA, 21 CFR Part 11, security and privacy regulatory issues, design and implementation of systems security, encryption, firewall configuration and review, cryptography, and security policy development. Ben is the author of Computer Security: 20 Things Every Employee Should Know and a contributing author to the Information Security Management Handbook.

Peter Stephenson, Ph.D., CISSP, CISM, FICAF, is Associate Program Director for the Master of Science in Information Assurance Program at Norwich University. The author of Investigating Computer-Related Crime, he has published several papers on information security issues in Information Systems Security.

Harold F. Tipton, CISSP, currently an independent consultant and Past-President of the International Information System Security Certification Consortium [(ISC)²,] was Director of Computer Security for Rockwell International Corporation for 15 years. He initiated the Rockwell computer and data security program in 1977 and then continued to administer, develop, enhance and expand the program to accommodate the control needs produced by technological advances until his retirement from Rockwell in 1994.

He has been a member of the Information Systems Security Association (ISSA) since 1982, was president of the Los Angeles Chapter in 1984, and president of the national organization of ISSA (1987 to 1989). He was added to the ISSA Hall of Fame and the ISSA Honor Role in 2000. He received the Computer Security Institute "Lifetime Achievement Award" in 1994 and the (ISC)², "Hal Tipton Award" in 2001. He was a member of the National Institute for Standards and Technology (NIST) Computer and Telecommunications Security Council and the National Research Council Secure Systems Study Committee (for the National Academy of Science).He has published several papers on information security issues in the Information Security Management Handbook, Data Security Management, Information Systems Security, and the National Academy of Sciences report, Computers at Risk.

He has been a speaker at all of the major information security conferences including: Computer Security Institute, the ISSA Annual Working Conference, the Computer Security Workshop, MIS Conferences, AIS Security for Space Operations, DOE Computer Security Conference, National Computer Security Conference, IIA Security Conference, EDPAA, UCCEL Security & Audit Users Conference, and Industrial Security Awareness Conference.

He has conducted and participated in information security seminars for (ISC)^2, Frost & Sullivan, UCI, CSULB, System Exchange Seminars and the Institute for International Research. He is currently the editor of the Information Security Management Handbook.

© Copyright 2005 Auerbach Publications.