



**The "Calm" in the Coming  
XML-Web Services Security Storm**

— Forum Systems

► **Hurwitz Report**



# The “Calm” in the Coming XML-Web Services Security Storm

— Forum Systems

## iii Executive Summary

This white paper describes some of the basic characteristics of Web Services, the associated security risks, and Forum Systems' approach to securing Web Services.

## 1 Overview

The idea that Web Services won't be deployed due to security concerns is hogwash.

## 1 Web Services Characteristics

The focus of Web Services is the data, not the presentation, as is often the case on the Web.

## 3 Securing Web Services

As with any new technology, Web Services must be evaluated for security requirements and capabilities.

## 4 Web Services Security Objectives

When it comes time to take the capabilities of granular policy, flexible security, and persistent security to individual XML transactions and documents, it is important to keep in mind some standard security principles.

## 6 Forum Systems

Forum Systems was founded specifically to address the unique security needs of Web Services.

## 7 The Hurwitz Take

Forum Systems leads the way in implementing Web Services security solutions that work today and are flexible enough to address new security requirements tomorrow.

A Hurwitz Group white paper written for:

Forum Systems  
4505 South Wasatch Boulevard  
Suite 330  
Salt Lake City, UT 84124  
[www.forumsys.com](http://www.forumsys.com)

Published by:  
Hurwitz Group, Inc.  
111 Speen Street, Framingham, MA 01701 ► Telephone: 508 872 3344 ► Fax: 508 872 3355  
Email: [info@hurwitz.com](mailto:info@hurwitz.com) ► Web: [www.hurwitz.com](http://www.hurwitz.com)

May 2002

© Copyright 2002, Hurwitz Group, Inc.

All rights reserved. No part of this report may be reproduced or stored in a retrieval system or transmitted in any form or by any means, without prior written permission.

## EXECUTIVE SUMMARY

---

Security is a critical requirement for any enterprise intending to harness the power of Web Services. Existing security models are insufficient for operation in a dynamic flexible environment like those being developed with Web Services. Forum Systems has created a security appliance that can meet the needs of security in the Web Services arena, with an eye toward flexible security that will grow with the needs of an organization.

This white paper describes some of the basic characteristics of Web Services, along with the associated security risks and the security principles that apply to this dynamic environment. Finally, it describes Forum Systems' approach to securing Web Services.

## Overview

Let's get real, quickly — the idea that Web Services won't be deployed due to security concerns is hogwash. Why? Because Web Services is poised to create tremendous business value for those who "dare" enter into the lair of real-time, dynamic transactions. And enterprises will swarm over this value proposition, all marching to the beat of XML, UDDI, and SOAP. But security is a crucial requirement for Web Services success. Woe to the enterprise that takes the leap and has no safety net. Now is the time to develop your security strategy for the coming Web Services wave.

## Web Services Characteristics

To understand what Web Services is, it is important first to understand what it isn't. It is not just a service provided over the Web. That is, the architecture is not the same as today's web environment, where the primary interaction is between the user and a web server. With Web Services, there is typically still interaction between user and web server, with the XML presentation language used to display web pages. But more importantly, there is the potential for complex interaction among multiple servers "behind the scenes." This is where services are defined, data is requested, and a transaction (or web page) is created, and then presented to the user or end entity. But the focus of Web Services is the data, not the presentation, as is often the case on the Web. This focus can be clarified within the following key characteristics of Web Services:

## Dynamic

Rather than taking months to create a detailed data model only to have it change during the review process, the Web Services infrastructure is designed to allow for dynamic data updates. This means, for example, that the CFO's change to a chart of accounts can be propagated among other systems or an upgraded inventory system can still interoperate with buyers' systems. In general, new data requirements can be easily accommodated, thus allowing for growth as the services are deployed and fuller functionality is added. This allows for real-time, on-the-fly modifications to

### Web Services Terminology

**XML** — The extensible markup language creates a way to define many different data formats so that platforms can interoperate. XML documents and transactions are made up of elements within a multi-level hierarchical structure.

**SOAP** — The simple object access protocol provides a network protocol for transport of Web Services documents.

**WSDL** — The Web Services description language provides a way to describe interfaces for Web Services.

**UDDI** — The universal description, discovery, and integration specification provides a registry for Web Services that can be searched for services and allows for dynamic updates.

data structures on the technology end that meet the needs of business partners or brokers on the business side. The old, static way of exchanging data by predefining data dictionaries and batch file formats for every partner or customer now becomes a fluid, dynamic transaction space where data can be transferred in real-time.

## Contextual

When two servers share information, they need some ability to understand the information in a way specific enough so that the systems can classify and use it. Web Services uses XML to add intelligence to the data by defining data fields along with their appropriate contextual relationships. So a field with the data value "MA" may have the XML tag <state> to provide enough information about the data to make it immediately useful. The combination of the data and the tag comprise an element. When these elements are nested to create a mailing address and an itemized bill, the end result is a hierarchical diagram that incorporates the data as well as the data labels (see Figure 1).

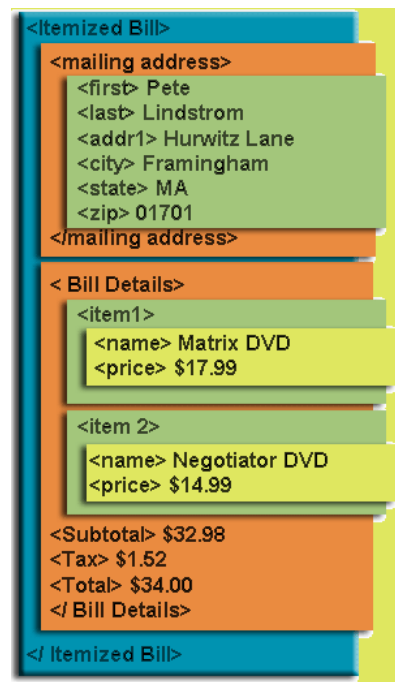


Figure 1. Hierarchical diagram of an itemized bill.

## Collaborative

Web Services has the capability to link together multiple sources of information or services into a single document or transaction. Riding on the existing web infrastructure, this

technology can extend the reach of the Web to higher value transactions. Its success is grounded in the capabilities that are already ubiquitous on the network. Web Services offers ubiquity on the application side — the ability to perform functions and transactions in a many-to-many relationship among business partners and independent providers, thus increasing the potential for revenue on the company side and satisfying demand on the customer side. For example, Web Services could combine a Primary Care Physician's diagnosis with a Specialist's treatment and a Pharmacist's prescription, as well as the prices for each, all submitted to an Insurance Provider for reimbursement.

## Result: Higher Reward (But Higher Risk)

The characteristics of Web Services almost predefine its success technically — it is a set of loosely integrated standards with flexibility built in that allows heterogeneous environments to work together. It creates the opportunity to offer up transactional data dynamically — a company can set rules associated with the data, verify in real-time when a new vendor makes an online request, and return the information back immediately with other processes spawning from them.

Along with the potential for high reward comes greater risk. The technical complexity of the environment can appear daunting when considering dynamic transactions among many different systems. This risk must be managed carefully so that an enterprise is leveraging the business value of new technology without being reckless or naïve.

## Securing Web Services

Web Services is a new technology that promises to provide significant benefits to an enterprise. Any new technology must be evaluated for security requirements and capabilities. Listed below are some of the capabilities that are useful in the Web Services space.

### Granular Policy

With Web Services, different elements or groups of elements are dynamically built into a document. This data can have multiple sources and destinations with varying security requirements. A granular security policy allows for mapping of the policy requirements to the transaction and applicable business partner. This allows for specific policies based on a user submission or a destination, and granularity with the ability to sign and encrypt individual data elements. A document that aggregates contract bids may need each bid individually signed and encrypted as it passes through a marketplace to protect the interests of participating competitors.

## Why SSL Is Insufficient

SSL has become the de-facto standard for ensuring security of Internet transactions. Consumers and businesses gain comfort from seeing the "little yellow lock" in the lower right corner of an SSL-enabled web browser. But that security is only skin deep; while SSL provides communications security, it can do nothing to protect the data stored on the server. Specifically, SSL is insufficient in the following ways:

- ▶ *SSL begins and terminates in concert with a communications session; there is no persistent security.*
- ▶ *SSL is point-to-point; it breaks down in a multi-point environment.*
- ▶ *SSL is not data-aware; it just encrypts everything that is there.*

SSL was never meant to handle the security needs of the Web Services environment.

## Flexible Security

As business models that leverage Web Services evolve, so must the security options. The highest degree of flexibility exists when a solution can secure at the element level. In some cases, such as credit card information in a bill that is processed by a third party, the ability to encrypt at the element level is already a necessity. In other scenarios, element-level encryption may not be important, but usage patterns could provide a way to generate new sources of income that makes it necessary. Perhaps a real-time inventory system today will lead to just-in-time replenishment from multiple suppliers tomorrow, turning a retailer into more of a dynamic broker. Flexible security means being able to grow with these new models, as new ways of aggregating data, working with brokers and other partners, and distributing information are created.

## Persistent Security

Like energy, data has two states — a "potential" resting state when data is stored in a queue or repository, and a "kinetic" motion state when data is being passed from point A to point B. Current security models secure the data in these two states differently, using access control for stored data in

databases and file systems and secure sockets-layer (SSL) for data transmissions. With Web Services, the ability to understand context can be applied from start to finish. Persistent security that accompanies the data can be useful for data that is stored in third-party repositories, then "unlocked" at some point in the future.

## Web Services Security Objectives

When it comes time to take the capabilities of granular policy, flexible security, and persistent security to individual XML transactions and documents, it is important to keep in mind some standard security principles. These principles are discussed below.

## Confidentiality

Credit card numbers are the obvious example when discussing confidentiality of data on the Internet. Certainly, protected information must be kept from the global threat of interception. This goes for basic transactions as well as intellectual property being shared among business partners. Confidentiality provides a basic level of “secrecy” when valuable data is being transmitted or stored in various places around the Internet.

## Integrity

Many types of errors can occur when transmitting data from source A to destination B. Completeness and accuracy are the ultimate goals when sharing data in a complex computing environment. Often, these goals are reached through complex programming algorithms or manual procedures. With Web Services, integrity can be provided using industry standards in a simple, efficient way.

## Authenticity

The problem of authenticity crops up when considering Web Services business models that charge based on the transaction, or information that must have a verifiable source. On the Internet, it is easy to provide information with no validity at all. Ensuring authenticity allows an enterprise to verify its sources, and allows the sources to receive appropriate compensation.

## Audit and Receipt

Web Services transactions are business transactions. It is necessary to audit the activity in two ways — first audit the activity that occurs on a technical level, such as configuration changes; and second, and more importantly, audit the transactions and provide a receipt to verify that the business transaction occurred.

### Web Services Security Standards

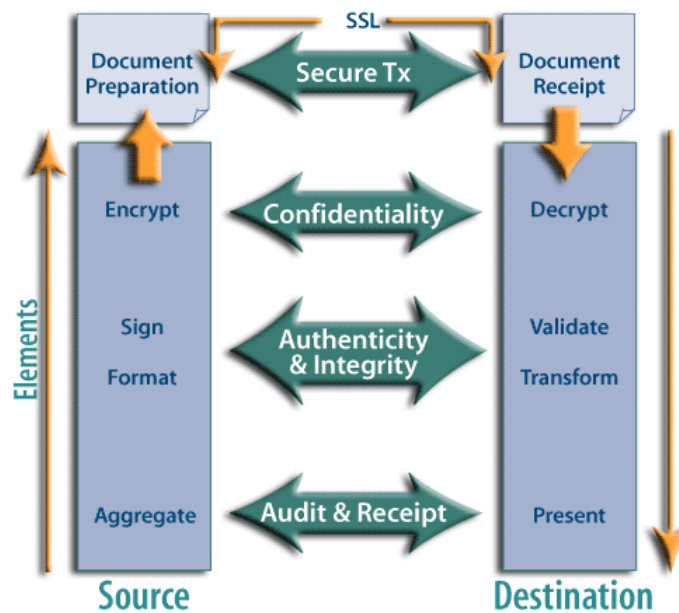
There are two key security standards (among others) that should be considered when deploying Web Services, both from joint working groups of the IETF and W3C:

- ▶ *XML Encryption provides a mechanism to ensure confidentiality of data within XML documents. It allows for encrypting the entire document, encrypting logical elements groups, or encrypting individual elements.*
- ▶ *XML Signature provides a mechanism to ensure integrity and authenticity of data within XML documents. XML Signature uses digital signatures and public key cryptography to ensure that different pieces of a document can be signed and verified.*

Other security standards exist that leverage the strengths of Web Services but can apply to both Web Services and legacy systems. These standards typically act on a session-level and include SAML (security assertion markup language) and XKMS (XML key management system).

## Securing the Web Services Transaction

With these principles, an XML transaction can be secured, as in Figure 2. Once a document is requested, the application server or Web Services aggregator will collect elements and documents from which to build the response. At that time, authenticity and integrity can be ensured by signing some combination of elements or the entire document. Confidentiality is guaranteed through encryption and also applied to some combination of elements, though not necessarily the same ones that were signed. Finally the document is prepared for transmission. Secure transmission within a session can be protected using SSL or IPSec for confidentiality of communications.



**Figure 2. Secure XML transaction.**

When the requesting system receives the document, it performs the opposite operations to decrypt encrypted elements, validate signatures for integrity and authenticity, and transform the document into data that is usable by the application. The final step is to provide audit and receipt capabilities, when the destination system returns a receipt to the source.

## Forum Systems

Forum Systems was founded specifically to address the unique security needs of Web Services. Its goal is to build Web Services security appliances that are easy to deploy and integrate into new or existing XML and Web Services applications. Forum characterizes its benefits as four cornerstones:

## End-to-End Data Integrity

Forum Systems appliances can digitally sign all or part of an XML/Web Services transaction such that its authenticity and integrity can be verified. Its solutions use cryptographic means to ensure that data is protected throughout the transaction process, which may span many different business partners and third-party services.

## Data-Level Confidentiality

Forum Systems provides flexible encryption for confidentiality of XML documents and transactions. With Web Services, data is dynamic; it can be parsed and segregated, then rebuilt or combined with other data to create new data and transactions. Some of the information must be kept confidential. Forum provides the capability to encrypt individual elements, grouped elements, or entire documents — meeting the needs of any Web Services deployment.

## XML Web Service Auditing

Forum Systems provides auditing capability to track transactions. This feature can be used to monitor activities for error control — to correct corrupted data, for example — or for business use, to track and bill a customer based on usage. In either case, the auditing capability is built in and includes a receipt mechanism for full notification.

## XML Data Processing

Forum Systems offloads processing of activities such as validation and transformation to free up processing of other components of a Web Services system. In this way, the Forum Systems appliance acts as a shipping point providing a final check of a document or transaction prior to its transmission, and as a receiving department verifying the proper receipt of information prior to forwarding it to its final destination.

## The Hurwitz Take

Forum Systems leads the way in implementing Web Services security solutions that work today and are flexible enough to address new security requirements tomorrow. Its security appliance provides an easy-to-implement means for securing Web Services. With a hardware based appliance, security operations are optimized for performance, while flexible configuration remains in the software. Its ability to provide a rich feature set allows for strong policy management, making Forum Systems a leader in this field.



## About Hurwitz Group

Hurwitz Group, an analyst, research, and consulting firm, is a recognized leader in identifying and articulating the business value of technology. Known for its real-world experience, consultative style, and pragmatic approach, Hurwitz Group provides strategic guidance to its clients by delivering analysis, market research, custom content, and consulting services. Clients include Global 2000, software, services, systems, and investment companies.