Doc. ID: ZTR-WS104-1
Released: June 20, 2002

zapthink
foundation report

XML AND WEB SERVICES SECURITY

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

XML AND WEB SERVICES SECURITY

JUNE 20, 2002

Analyst: Jason Bloomberg

Abstract

Security is the immediate roadblock facing widespread implementation of Web Services technologies across the enterprise. As a result, many software vendors are throwing their hat into the XML and Web Services security ring, offering a broad and confusing number of solutions to a variety of real and perceived problems. However, much of this effort amounts to jostling for defensible market positioning ahead of a solid demand for enterprise-class XML and Web Security products and services. As a result, ZapThink believes that the emerging market for XML and Web Services security solutions will be characterized by a period of turbulence, as companies struggle to clarify their messages and shake the kinks out of their product offerings.

Key Points:

✦ Market Overview

- Security is the major roadblock to Web Services adoption.
- There is tremendous confusion over implementing secure Web Services due to conflicting, overlapping, and incomplete security offerings.

✦ Facts & Figures

- The XML and Web Services security market will reach $4.4 billion in 2006, growing over 300% annually.

✦ Analysis

- The lack of robust security and manageability solutions inhibits the ability for companies to effectively utilize Web Services for integration with business partners.
- Emergent startups have remarkably robust XML/WS security solutions due to adequate funding, solid business models, seasoned management teams, and high quality engineering staff.
- Companies that have deep technical knowledge of application level security coupled with a solid customer base will be best poised for success in the XML and Web Services security space.

✦ Future Trends

- Demand for XML/WS security solutions will spike within the next 12 months.
- Web Services will play limited role in transactional environments until 2003.
- By 2006, most security products from existing vendors will support or provide XML and/or Web Services security.

✦ Decision Points

- Securing a company's Web Services outside of the context of an overall security strategy provides a false sense of security.
- Enterprises must institute policies that apply to their entire extended enterprise and administer that security in a hierarchical fashion.
- Next-generation firewalls must be capable of looking at and securing the content of XML streams .

Table of Contents

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

I. Report Scope

Locking 19 entrances to a building does not provide security if there are 20 entrances.

Fundamentally, this report covers the overlap of two large areas of general interest: IT security, and the combination of XML and Web Services. Therefore, the scope of this report must be put into the context of each of these areas. IT security solutions address how to mitigate risks in the enterprise at all levels. Locking 19 entrances to a building does not provide security if there are 20 entrances, and the same is true for IT. Therefore, discussions of security must cover all aspects of security, including human, physical, network, and application security.

Within the XML and Web Services world there are two main challenges: how to secure Web Services (and in general, any kind of XML message), and how XML and Web Services-based solutions can provide security to the enterprise. Each challenge must also be placed within the overall context of IT security; after all, a hacker doesn't care if a particular vulnerability is Web Service related.

This ZapThink report, *XML and Web Services Security*, covers the various products and services on the market today that focus on meeting the requirements for providing security to enterprises that use Web Services, as well as products and services that use Web Services to provide security to enterprises. The report identifies benefits, challenges, key market drivers, and determines the sizing and growth of the market for these products and services.

This report covers:

- The context for XML and Web Services security within both the Web Services model for distributed computing and the IT security landscape
- The technology landscape for XML and Web Services security, including:
  - The layers of IT security
  - Precursors to XML and Web Services security, including PKI, Kerberos, and SSL
  - Key ongoing efforts in XML security, including XML signatures, XML encryption, and XKMS
  - Key ongoing efforts in Web Services security, including SAML and WS-Security
- Segmentation of the XML and Web Services security market
- The current state of the XML and Web Services security market, including the advantages and disadvantages of various approaches to security, and the return on investment (ROI) for implementation of the various products and services in this space
- Business and technology trends in the XML and Web Services security market, including:
  - ZapThink's predictions for market growth
  - inhibitors to growth
  - Market sizing and consolidation
- Profiles of vendors offering XML and Web Services security solutions.

This report does not cover:

- Global identity services, namely the Liberty Alliance and Microsoft Passport and TrustBridge. These will be covered in ZapThink's upcoming *Global Identity Services* report.
- Digital Rights Management products and services.
- Directory services (LDAP, Active Directory, etc.)
- Web Service infrastructure and management vendors who do not have a specific security offering. Web Service infrastructure and management

will be covered in the upcoming ZapThink Report *Web Services Infrastructure and Management*.

ZapThink hopes that this report will provide a fundamental understanding of the security issues surrounding XML and Web Services, and will provide a clear picture of the current state as well as the future of this dynamic, emerging market.

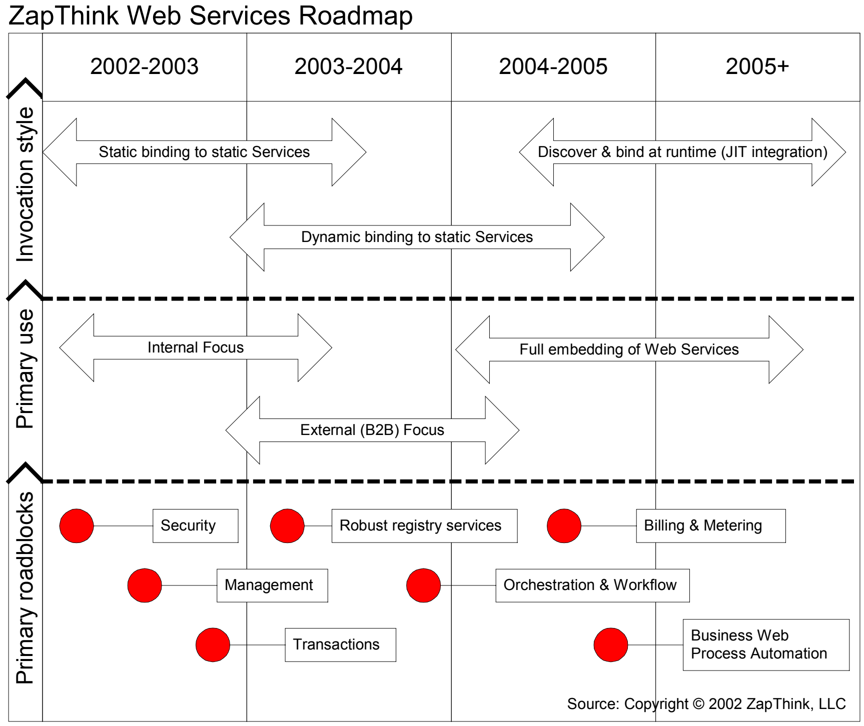
II. Context for Security in the Web Services Model

ZapThink believes that Web Services are an evolutionary step in the development of distributed computing characterized by open standards, loose coupling, and dynamic description and discovery. (For our complete definition of Web Services, please see the ZapThink report *The Pros & Cons of Web Services*. For a broad look at the Web Services market, refer to the ZapThink report *Web Services Technologies & Trends*). Nevertheless, to say that Web Services are evolutionary rather than revolutionary is not meant to diminish their importance. It is important to point out, however, that the *promise* of Web Services is still quite different from today's reality. Therefore, it is a goal of this report—and truthfully, part of ZapThink's mission—to provide our subscribers with a clear roadmap, that places today's implementations of Web Services in context, and then provides a rational picture of where we feel Web Services will take distributed computing and the whole of IT in the future.

2.1 The ZapThink Web Services Roadmap

To understand the context of security in the discussion of Web Services, it is important to understand how Web Services will affect distributed computing over the next five or more years, as shown in the ZapThink Web Services Roadmap in Figure 2.1.

Figure 2.1: ZapThink Web Services Roadmap



The primary use for Web Services today is for internal integration.

The primary use for Web Services today is for internal integration, where the invocation style is for Web Service consumers to bind to static Web Services at design time – basically, simply replacing existing, proprietary RPC protocols with open standards-based protocols. However, in the 2003-2004 timeframe, ZapThink sees more companies taking advantage of Web Services' dynamic binding capabilities. This invocation style will generally coincide with a greater use of Web Services outside the firewall between business partners. Finally, in the 2004-2005 timeframe, we believe that companies will be capable of discovering and binding to Web Services at runtime, in a mechanism known as "Just-in-time (JIT) integration". JIT integration enables enterprise software vendors to rearchitect their suites as loosely coupled collections of Web Services. At that time Web Services will be fully embedded into the software development process.

Decision Point

The next roadblock on the path to Web Services adoption is security. Security is today's key enabler for Web Services.

However, this vision of the future of Web Services is not without its roadblocks. In fact, there are seven major roadblocks that promise to inhibit or even derail progress as shown on the Web Services roadmap. Adoption of Web Services will progress up to the point that the next roadblock begins to impact the further acceptance of Web Services. Only when the relevant issues are resolved and the roadblock disappears will Web Services continue on its path toward widespread adoption.

2.2 Security: The Key Enabler for Web Services

When Web Services is used for static communication between two internal, controlled systems, security is not a major issue since existing security

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

techniques (firewalls, SSL, etc.) usually suffice to meet an organization's risk mitigation requirements. However, for any application of Web Services that provides access outside the corporate firewall, or intra-enterprise applications where the Web Services consumer is not fully controlled by the IT department, security issues are clearly the first and foremost issue that IT must resolve. Therefore, security is today's key enabler and roadblock for Web Services.

Of all the concerns on today's executives' plates, security is the most complex. Security is essentially the mitigation of risk, and risk is something every company has and wants to rid itself of. Risk, however, is a slippery concept, because it deals in possibilities: the possibility of a break-in, or a virus, or a loss of confidentiality, for example. Lowering risk means lowering the probability that such future events might occur; therefore, the ROI on security investments can only be expressed in terms of how much a security failure would have cost if it had happened, even though it hasn't. Such ROI calculations are particularly difficult, because the greatest risk is one that is completely unexpected—one that by definition wasn't predictable.

This report must be placed into the context of an overall security strategy. Simply securing all of a company's Web Services alone only provides a false sense of security.

The other main reason that security is such a complex issue is the "twenty doors" problem. Simply securing one part of a system, or a network, or a building complex provides only false security. All possible risks must be considered, including those unpredictable ones that are potentially the most costly. Therefore, it is critical that this report be placed into the context of an overall security strategy. If a company increases its risk by not securing its Web Services, then there's no question that this door must be closed and locked. However, simply securing all of a company's Web Services alone only provides a false sense of security.

Therefore, while this report necessarily must focus on securing XML and Web Services, all of the discussions herein must be considered within the context of the greater security landscape. The next five sections place the discussion of securing XML and Web Services in the greater context of IT security and related markets, in order to provide readers with the "big picture" of IT security as it pertains to XML and Web Services.

2.3 Context: Security Products & Services

From the broadest perspective, security encapsulates three major categories: human, physical, and IT. The human side of security involves hiring policies, training, password and physical key policies, and various cultural issues. Physical security includes lock and key systems, surveillance technologies, alarms, and other such protection systems. IT security is the final category and is concerned with securing computer hardware and software, networks, devices, etc.

The IT security market can be segmented as follows:

- Hardware security solutions, divided into:
  - Firewall appliances
  - Other security appliances (secure routers, etc.)
  - Authentication hardware
- Professional services solutions for security.
  - Protection services
  - Security audit
  - Intrusion detection services
- Software security solutions, segmented into:
  - *Authentication, authorization, and administration* ("3A").

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

- Encryption, a smaller market that often overlaps with 3A. For the purposes of this report, therefore, we will consider encryption to be part of the 3A IT security services segment.
- Software that runs on firewalls and other security devices.
- Anti-virus software.

Most XML and Web Services security offerings on the market fall within the 3A (authentication, authorization, and administration) segment of the IT security market.

Most XML and Web Services security offerings on the market fall within the 3A segment, but there are several important exceptions to this generalization. We will consider each of them:

- Many hardware firewall vendors are developing support for XML and Web Services. In addition, there are vendors such as **Sarvega** and **Forum Systems** who are developing new kinds of hardware network appliances in order to provide different levels of XML and Web Services security. ZapThink's upcoming report on *XML and Web Services Network Appliances* will provide detailed coverage of these segments, including both the firewall hardware and the software that runs on them.

- Anti-virus software does not protect XML or Web Services *per se*. However, major anti-virus software vendors, including **Symantec** and **McAfee**, have been offering their desktop anti-virus products via Web-based service interfaces for a while now. Both companies are considering moving their currently proprietary service infrastructures to the open Web Services standards, which would make them *bona fide* Web Services. However, providing anti-virus software via a Web Service is an example of an application of Web Services for security, rather than an example of securing XML and Web Services.

Vendor Focus

Forum Systems
Sarvega
Symantec
McAfee
TruSecure
ISS
IBM
EDS

- IT security professional service providers like **TruSecure**, **ISS**, and divisions of **IBM** and **EDS** offer a range of professional services options to their customers, which increasingly include Web Services security professional services. For these companies, Web Services initiatives are just one more kind of risky project their customers might undertake. Each service provider is thus updating their procedures to include the proper advice, configuration management, intrusion detection, and auditing services to reflect the new protocols and programming techniques that Web Services introduce to the enterprise. Broadly speaking, therefore, IT security professional services are not being fundamentally changed by Web Services; instead, securing Web Services is just one more wrinkle in the complex set of protective services such companies offer their customers.

Most 3A products will be Web Service-enabled by 2005.

So, while this report will touch upon the above segments, the core of the story about XML and Web Services security falls within the 3A security software segment. As we will show in section 6, the interesting story is not just how Web Services lead to new kinds of products within the 3A segment, but also how Web Services technologies will gradually propagate throughout the entire segment, where most 3A products will be Web Service-enabled by 2006.

Vendor Focus

Bowstreet
Cape Clear
Iona
Systinet
AmberPoint
Flamenco Networks
Microsoft
IBM
HP

2.4 Context: Web Services Management and Infrastructure Products

There are several startups and emerging companies fighting over the Web Services infrastructure space, including **Bowstreet**, **Cape Clear**, **Iona**, and **Systinet**, to name just a few. Each of these companies' product offerings includes security functionality in some form. In addition, there is now a burgeoning Web Services management segment, populated by the likes of **AmberPoint** and **Flamenco Networks.** Naturally, **Microsoft** also provides

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

management capabilities within its .Net framework, and **IBM** is revamping its Tivoli system management product to provide robust Web Services management capabilities. It is also likely that **HP** is doing something similar with OpenView, although at this time HP's strategy is still up in the air due to their recent merger with Compaq. Managing the security aspects of Web Services are part of what each vendor in this category does, as well.

As shown in figure 2.1, Web Services management is the next major roadblock to Web Services adoption after Web Services security. It follows, therefore, that many companies looking to provide solutions in the Web Services management space must also resolve many of the issues surrounding Web Services security. For the purposes of this report, however, we will only be considering Web Services management and infrastructure vendors insofar as their XML and Web Services security products are important to the market. ZapThink covers the emerging Web Services Management market in its *Web Services Infrastructure and Management* report.

2.5 Context: Global Identity Services

☆ Vendor Focus

Microsoft
Sun Microsystems

Another market segment closely associated with the XML and Web Services security space is the global identity services segment, populated most notably by **Microsoft**'s Passport and recently announced TrustBridge initiatives, and the **Sun Microsystems**-led Liberty Alliance. Each party in this brewing feud promises to offer both consumers and businesses a universally usable way of managing user identity and authentication mechanisms. However, since the Liberty alliance has yet to announce many aspects of their recommendations, Zapthink is publishing this report a few months too early to cover this segment in depth. Therefore, ZapThink will cover this area in its future report on *Global Identity Services*.

2.6 Context: Digital Rights Management Technologies

☆ Vendor Focus

ContentGuard
Microsoft
InterTrust

The Digital Rights Management (DRM) space is also closely related to the XML and Web Services security space as well. Originally, DRM applied primarily to media assets (audio and video), but now increasingly applies to XML formatted content, as well. Vendors are using DRM technologies to apply usage metering and permission systems for the use of Web Services as well as the protection of the XML content from inappropriate usage or distribution. A primary source of overlap between the DRM and XML security areas results from the development of the Extensible Rights Markup Language (XrML), developed by **ContentGuard** and championed primarily by **Microsoft**.

XrML provides a universal method for specifying and managing rights and conditions associated with digital content as well as services, including Web Services. At this time, XrML is the primary rights language being used in working DRM solutions, and in particular, those from Microsoft. The reason that XrML overlaps XML and Web Services security is largely a byproduct of its complexity. In fact, XrML has been growing increasingly more complex as it has moved to the current version 2.0. For example, XrML 1.3 defined 18 different rights that can be conferred on content. XrML 2.0 expands this number to 24. The newly added rights include those that have to do with the issuance of digital signatures and the read/write/execute rights that are a key part of authorization mechanisms.

XrML-based DRM technologies will not form an independent market by themselves. Instead, the DRM space will eventually merge with the work in XML and Web Services security.

While Digital signatures are an important part of the XML security story, the question remains whether it is really necessary to define digital signatures as part of a DRM specification. The fact that XrML is overly complex, has relatively few supporters, and overlaps the more widely accepted efforts in the XML security space leads us to predict that XrML-based DRM technologies will not

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

form an independent market by themselves. It is more likely that with the assistance of DRM leaders like **InterTrust** and others, the DRM space will eventually merge with the ongoing work in XML and Web Services security.

2.7 Context: Directory Servers

Vendor Focus

Novell
Sun Microsystems
Microsoft

Ever since the unwieldy X.500 directory standard was supplanted by its simpler and more flexible cousin LDAP, the directory server market has taken off. Leaders in the LDAP directory server space include **Novell** and **Sun Microsystems**, while **Microsoft** provides an alternative through its more feature-rich Active Directory. Directory servers appear at the heart of most enterprises, managing hierarchical repositories of information on people and physical resources. LDAP is also capable of working with security constructs like digital signatures and certificates, making directory servers the favored user access management tool across enterprises worldwide.

Naturally, most of the products that this report covers are capable of accessing LDAP and/or Active Directory servers, either as a core part of their solution, or as one of the supported interfaces. The directory server story, therefore, is an important part of the XML and Web Services security story. Nevertheless, the directory server market is by no means XML-specific, and is generally considered to be a market in its own right.

III. Technology Landscape

To understand the XML and Web Services security market, it is important to have a high-level understanding of software security, as well as the precursor technologies that provide the basis for XML and Web Services security. It is also important to understand what makes XML and Web Services security different than typical Web or network security.

Application level security focuses on which services users can request, based upon what roles those users have in the system, and also provides for the confidentiality and integrity of the transmitted data.

There are two basic types of IT security:

- *Perimeter network security* focuses on which machines are on what networks, what protocols will be allowed to pass across these networks, and which TCP/IP ports on each device will allow traffic from particular sources.
- *Application level security*, on the other hand, focuses on which services users can request, based upon what roles those users have in the system. Application security also provides for the confidentiality and integrity of the transmitted data, both in transit as well as in storage.

XML and Web Services security is primarily concerned with application level security, because Web Services send messages through firewalls typically using TCP/IP ports 80 (standard Web traffic) and 443 (SSL-secured Web traffic), thereby bypassing typical firewall restrictions. Earlier RPC (remote procedure call) technologies like CORBA and Java RMI communicate on different ports, which ironically has limited the usefulness of those protocols. Public-key encryption rollouts have also been affected by this problem, because firewalls typically block LDAP directory lookups, which use port 389.

Decision Point

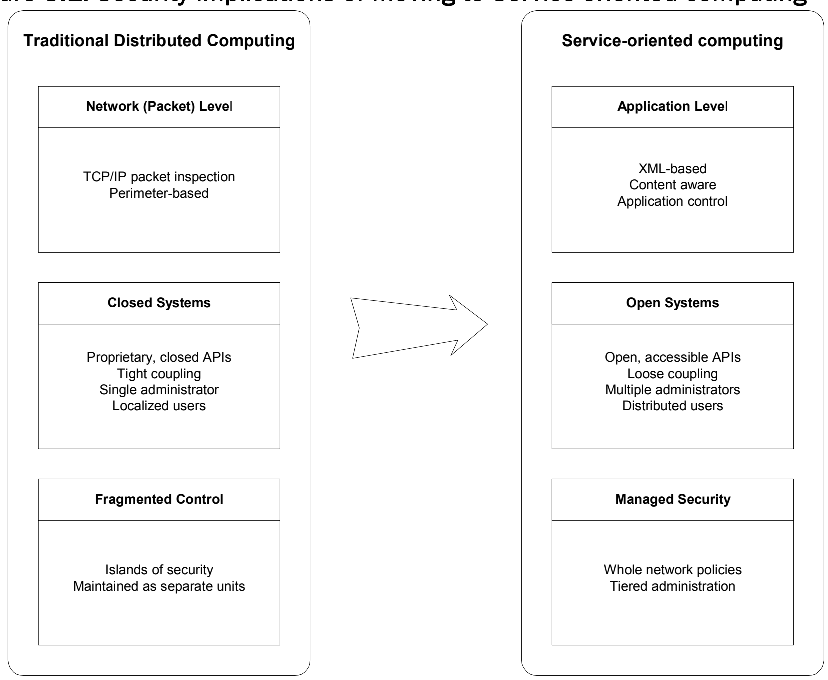
Next-generation firewalls must be capable of looking at the content of XML streams, and the security mechanisms for such data must be part of that content.

Therefore, XML and Web Services circumvent the security offered by firewalls that simply deny network traffic based on particular TCP/IP port numbers. Next-generation firewalls must be capable of looking at the content of XML streams, and the security mechanisms for such data must be part of that content. The security mechanism must be sophisticated enough, however, to verify the security of the message without compromising that security. This issue is fundamental to securing XML and Web Services.

3.1 XML security and the shift to Service-oriented computing

The shift from focusing on packet level network security to application level security that is aware of the contents of messages is one of the many changes facing an enterprise as it implements Web Services. As shown in figure 3.1, there are many levels of change facing IT in the enterprise, and each type of change has security implications associated with it. In addition to the need for firewalls to be application and content aware, there are changes at the system level as closed, proprietary systems give way to open, loosely coupled systems. Closed systems are relatively straightforward to secure; an administrators only needs to set up the users and their privileges, and the work is mostly complete.

Figure 3.1: Security implications of moving to Service-oriented computing



Source: Copyright © 2002 ZapThink, LLC

Decision Point

Enterprises must institute policies that apply to their entire enterprise network (including participants invited from outside), and administer that security in a hierarchical fashion.

Securing open, loosely coupled systems requires a much more sophisticated security approach, involving multiple administrators that support distributed users. Different systems now have different policies and possibly different security mechanisms. As a result, administrators must manage security much more actively than was necessary in the closed model.

Traditional distributed computing security was modeled by *islands of security*, which describe systems and users on isolated networks or subnetworks. The network acted as an island, with its own perimeter security, but users within the network were considered to be trusted. This “trusted vs. untrusted” dichotomy breaks down in a Service-oriented model, because users can access Services located on systems across one or more enterprises. The concept of trusted groups no longer has meaning; instead, enterprises must institute policies that apply to their entire enterprise network (including participants invited from outside), and administer that security in a tiered, or hierarchical fashion.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Departments or other organizational groups may then have their own administrators, but those administrators may in turn be administered by a more senior admin at a higher level within the enterprise.

3.2 Principles of Application Security

In order to fully understand how security can be provided and managed in the Service-oriented enterprise, it is important to first understand the principles of application level security.

TCP/IP forms the basis for all Internet-based communications. It allows systems to send information from one computer to another through a variety of intermediate computers and separate networks before it reaches its destination. TCP/IP's great flexibility has led to its worldwide acceptance as the basic Internet and intranet communications protocol, but the fact that it allows information to pass through intermediate computers makes it possible for a third party to interfere with communications. Securing these points of attack form the basis for application level security in a TCP/IP environment:

- *Eavesdropping* -- Information remains intact, but an unauthorized person compromises its privacy. For example, someone could learn a credit card number, record a sensitive conversation, or intercept classified information.

- *Tampering* – An unauthorized person changes or replaces information in transit to the recipient. For example, someone could alter an order for goods or change a person's resume.

- *Impersonation* -- Information goes to a person who poses as the intended recipient. Impersonation can take two forms:

    - *Spoofing* -- A person can pretend to be someone else. For example, a person can pretend to have someone else's email address, or a computer can identify itself as a Web site it is not.

    - *Misrepresentation* -- A person or organization can misrepresent itself. For example, a site might pretend to be a bookstore when it is really just a site that takes credit-card payments but never sends any goods.

These potential security breaches lead to the following requirements for application level security.

3.2.1 Application level security requirements

Application level security contains five basic requirements, expressed in terms of the messages sent between parties. Such messages include any kind of communication between the sender (party who wishes to access an application) and the recipient (the application itself). The five requirements for application level security are:

- *Authentication*, The recipient of the message must be able to confirm the identity of the sender of the message.
- *Authorization*. The sender of a message must be authorized to send the message.
- *Confidentiality*. The contents of messages must not be available to unauthorized parties.

- *Data integrity*. The recipient of a message must be able to guarantee that a message hasn't been tampered with in transit.
- *Nonrepudiation*. The sender must be able to guarantee that the recipient received the message, including the time the message was sent and the fact the recipient received only a single copy.

This report will take each of these five requirements in turn, and explain the issues involved in satisfying each requirement within an XML and Web Services security context.

3.2.2 Authentication

A system provides authentication if it allows access only to those users who can provide the proper identity credentials. Systems commonly handle authentication with a user ID and password as the identity credentials. However, user ID/password security offers only a relatively low level of protection. Other authentication schemes use software tokens as identity credentials. PKI, for example, uses a digital certificate as the identity credential, and Kerberos uses a ticket. Only the person who owns a security token, whether it's a certificate or a ticket, can use the token.

There must be an identification mechanism outside of the software system that provides the initial confirmation of identity.

One of the biggest issues with authentication is the bootstrap problem: to obtain a security token a user must prove their identity to the issuer of such tokens, but how can they do that without a token? The answer is that there must be an identification mechanism outside of the software system that provides the initial confirmation of identity—a user can only get their first security token by providing a driver's license to the token issuer, for example. Once the user obtains one security token, then it is possible to use that token to provide authentication to multiple systems.

3.2.3 Authorization and Access Control

Authorization determines whether a user is allowed to perform the functions it requests or access requested data. Authorization is particularly important because of the need for tiered security administration in Service-oriented environments, where security administrators delegate their administrative functions to other administrators. At the simplest level, systems handle authorization with access control lists (ACLs) that list which users are entitled to perform certain operations (e.g., read, write, delete) on particular resources. However, ACLs are generally insufficient to handle the real-world security policies required at many enterprises, because Web Services provide programmatic interfaces that are difficult to monitor for suspicious activity. Take, for example, an HR application that has a Web Service interface. A request for Mary's salary would raise immediate suspicion from a human HR representative, but access to an improperly protected SOAP interface to the HR system would be more difficult to detect.

The ability to provide and administer authorization across multiple systems is a difficult problem that many of the companies covered in this report are trying to solve.

This situation is even more complex when multiple, heterogeneous systems are involved, either within an enterprise or across two or more companies. Every company will likely have its own security policies, in addition to its own authorization technology. Therefore, the ability to provide and administer authorization across multiple systems is a difficult problem that many of the vendors covered in this report are trying to solve.

3.2.4 Confidentiality

Confidentiality means that an unauthorized person cannot view or interfere with a communication between two parties. Trust infrastructures like the *Public Key Infrastructure* (PKI) and Kerberos use encryption to ensure that messages are kept confidential. PKI in particular can use encryption to protect the confidentiality of data both in transit and in storage. Virtual Private Networks (VPNs) and *Secure Sockets Layer* (SSL) can protect the confidentiality of messages between two endpoints, but neither secures the data in storage or across intermediaries, because both SSL and VPNs are point-to-point techniques. Therefore, an SSL-encrypted message, for example, would have to be unencrypted at an intermediary, which opens a security hole.

Intermediaries are especially important in the context of Web Services, because the SOAP protocol is designed to support intermediaries that can forward or reroute SOAP messages.

The problem of intermediaries is especially important in the context of XML and Web Services, because the SOAP protocol is designed to support one or more intermediaries that can forward or reroute SOAP messages based upon information either in the SOAP header or the HTTP header. Therefore, there must be a way for the intermediary to read the part of the message that tells it what to do, without compromising the confidential payload of the message. However, technologies such as SSL prevent the effective functioning of these intermediaries.

3.2.5 Data Integrity

Data integrity comprises two requirements: first, the data received must be the same as the data sent. In other words, the message did not change in transit, either by mistake or on purpose. The second requirement for data integrity is that at any time in the future, it is possible to prove whether different copies of the same document are in fact identical.

PKI, for example, uses a technique called a *message digest* or *one-way hashing* to ensure data integrity. Hashing is an algorithm that takes any message and calculates a much shorter string of characters, known as the message digest, in such a way that performing the same algorithm on the same message again always yields the same result. The subsequent chance of two different messages yielding the same message digest is astronomically small. If the message digest is the same when a message is sent and when it is received, and the digest itself was kept secure, then data integrity is guaranteed. Likewise, a message can be hashed at two different points in time to determine if it has been altered.

3.2.6 Non-Repudiation

When secure messages are sent, the recipient often requires that the sender can't repudiate the message, or claim that the message wasn't sent at particular date and time. Likewise, a sender would like to guarantee that a given message was received. The most common way to provide non-repudiation is through the use of digital signatures. With digital signature technology, senders can both provide evidence that a document is valid while simultaneously logging the message transactions into signed audit logs. Once an audit log has been signed it cannot be surreptitiously modified.

3.3 IT Security Fundamentals

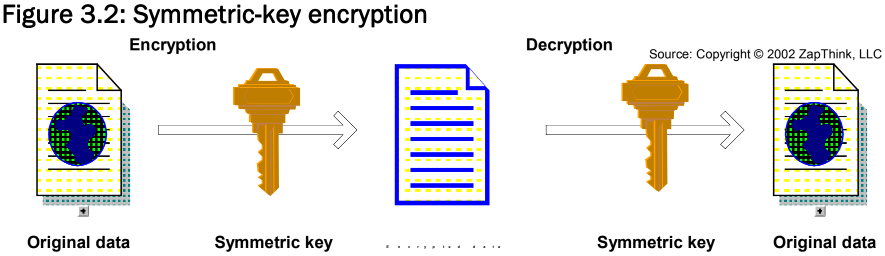
It is also helpful and important to understand the fundamentals of the existing application level security technologies in place. Much of this technology is based on a security construct known as a "*key*". A key is a string of characters that acts as a code that can be used to encrypt messages, and then decrypt them at the other end. There are two basic approaches to key encryption: *symmetric* and *asymmetric* key encryption. Symmetric key techniques, including Kerberos and SSL, use a specially generated secret key that is used as a shared secret between two communicating parties. If both parties have the same key, the message is secure. PKI, however, uses asymmetric key encryption, where there is a *public key* and a *private key*. The private key is used to encrypt a message, and the public key decrypts the message. Therefore, private keys are secrets kept with their owners, while public keys are truly public.

3.3.1 Encryption and Decryption
Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. *Decryption* is the process of transforming encrypted information so that it is intelligible again. A *cryptographic algorithm* is a mathematical function used either for encryption or decryption. In most cases, systems use two related algorithms: one for encryption and the other for decryption.

With most modern cryptography, the ability to keep encrypted information secret is not based on the cryptographic algorithm itself, which is widely known, but on the key that a system must use in conjunction with the algorithm to produce an encrypted result or to decrypt previously encrypted information. Decryption with the correct key is a simple, quick process. Decryption without the correct key, however, should be impossible – in reality, requiring significant effort.

3.3.2 Symmetric-Key Encryption
The simplest form of key encryption is known as *symmetric-key encryption*. With this form of encryption, systems can calculate the encryption key from the decryption key and vice versa. With most symmetric algorithms, systems use the same key for both encryption and decryption, as shown in Figure 3.2.

Figure 3.2: Symmetric-key encryption



Symmetric-key encryption can be highly efficient and also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. As long as both parties keep the symmetric key secret, each party can be sure that it is communicating with the other.

Symmetric-key encryption is only effective, therefore, if both parties keep the symmetric key secret. If anyone else discovers the key, then the communication loses both confidentiality and authentication. A person with an unauthorized symmetric key not only can decrypt messages sent with that key, but also can encrypt new messages and send them, thus spoofing either party.

3.3.3 Public-Key Encryption
Public-key encryption is a form of *asymmetric* encryption, because it uses a pair of keys—a public key and a private key—for each entity that wants to authenticate its identity, sign data, or encrypt data. The entity publishes its public key, but keeps the corresponding private key secret. If a system uses a public key to encrypt data, that data can only be decrypted with the corresponding private key. Figure 3.3 illustrates public-key encryption.

Figure 3.3: Public-key encryption

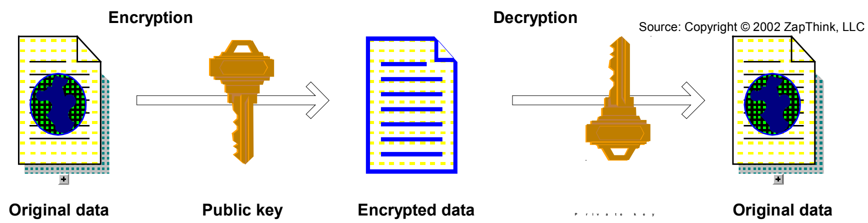


Figure 3.3 shows that a user can freely distribute a public key, but only that user will be able to read data encrypted using this key. In general, to send encrypted data to someone, the user encrypts the data with the intended recipient's public key, and the person receiving the encrypted data decrypts it with their corresponding private key. Compared with symmetric-key encryption, public-key encryption requires more computing effort. SSL solves this efficiency limitation by using public-key encryption to send a symmetric key, which it then uses to encrypt additional data.

The process illustrated in figure 3.3 works in reverse when the goal is to digitally sign data. If a user encrypts a message with their private key, then anyone can decrypt it, but recipients can confirm that the user who sent the message actually signed it. Such digital signing also confirms that an unauthorized party hasn't tampered with the message since it was signed.

3.3.4 Digital Signatures

Encryption and decryption address the problem of eavesdropping, but do not alone address the issues of tampering and impersonation. Tamper detection and related authentication techniques rely on message digests. A message digest has the following characteristics:

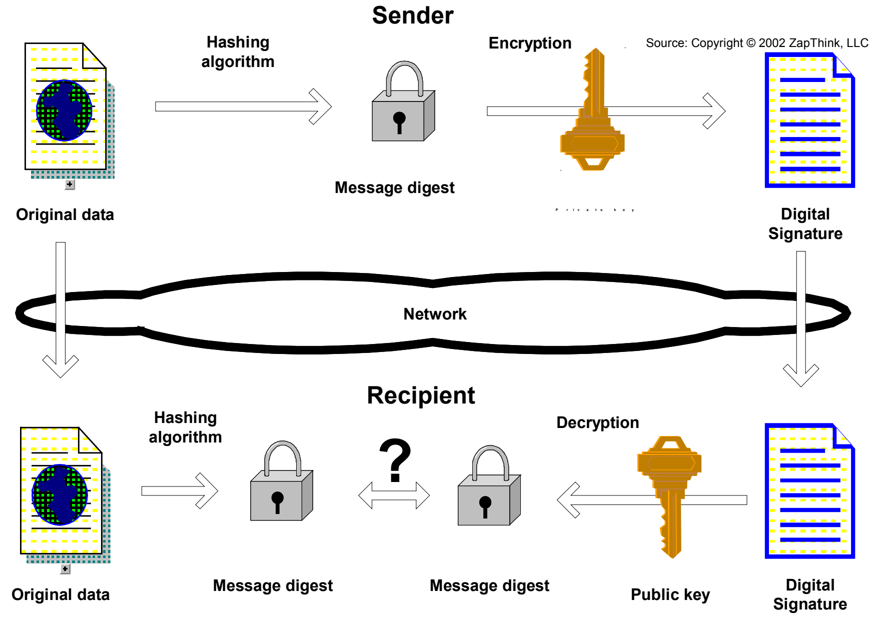
- The value of the digest is unique for the hashed data. Any change in the data, such as deleting or altering a single character, results in a different value.
- No one can deduce the content of the hashed data from the digest, which is why this technique is called "one-way."

As pointed out above, it is possible to use a private key for encryption and the corresponding public key for decryption in order to digitally sign data. In practice, however, signing software first creates a message digest of the data, and then uses the private key to encrypt the digest. The encrypted digest is called a *digital signature*.

Figure 3.4 illustrates how digital signatures validate the integrity of signed data.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Figure 3.4: Using a digital signature to validate data integrity



In the above diagram, the recipient receives the original data and the digital signature. To validate the integrity of the data, the receiving software first decrypts the digest with the signer's public key. It then uses the same hashing algorithm that generated the original digest to generate a new message digest of the same data, which it compares to the original digest. If the two digests match, the integrity of the message is confirmed, and the recipient can be sure that the public key the recipient has corresponds to the private key that the sender used to create the digital signature. Confirming the actual identity of the signer, however, requires the additional step of confirming that the public key really belongs to a particular person or other entity, which requires a certificate.

3.3.5 Digital certificates

A *certificate* is an electronic document that can identify an individual, a server, a company, or any other entity. The certificate associates that identity with a public key. Like a driver's license, a certificate provides generally recognizable proof of a person's identity. Public-key cryptography uses certificates to address the problem of impersonation.

The issuing of certificates works pretty much the same way as driver's licenses and other familiar forms of identification. *Certificate authorities* (CAs) are entities (typically companies, but they could also be internal to enterprises) that validate identities and issue certificates. The certificate the CA issues binds a particular public key to the name of the entity the certificate identifies (such as the name of a person or a server). Certificates help prevent the use of fake public keys. Only the public key that the certificate certifies will work with the corresponding private key that the entity the certificate identifies possesses.

In addition to a public key, a certificate always includes the digital signature of the issuing CA, which allows the certificate to function as a "letter of introduction" for users who know and trust the CA but aren't familiar with the entity the certificate identifies.

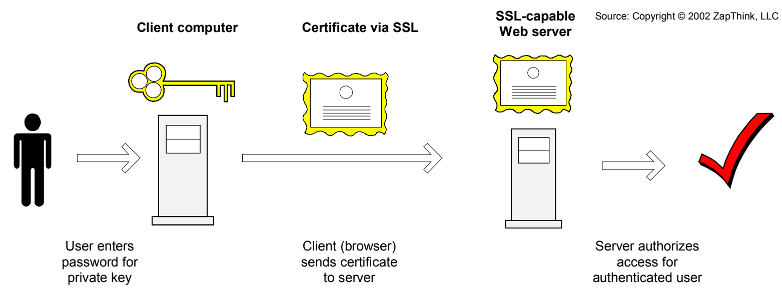
Certificate-based authentication is more secure than simple password-based authentication, because the user must have the private key as well as the password.

3.3.6 Authentication with certificates

Digital certificates support several different types of authentication, including the ability to authenticate the client (for example, a person using a browser), or a server (for example, an SSL-secured Web site). In addition, certificates authenticate digital signatures. In addition to authentication, digital signatures ensure nonrepudiation: a digital signature makes it difficult for the signer to claim later not to have sent the signed message.

Figure 3.5 shows how client authentication works using certificates and SSL. To authenticate a user to a server, a client digitally signs a randomly generated piece of data and sends both the certificate and the signed data across the network. The server authenticates the user's identity based on the information in the digital signature. This example assumes that the user has already decided to trust the server and has requested a resource (typically a Web page), and that the server has requested client authentication in order to decide whether to grant access to the requested resource.

Figure 3.5: Using a certificate to authenticate a client to a server



Certificate-based authentication is more secure than simple password-based authentication, because the user must have the private key as well as the password. However, neither password-based authentication nor certificate-based authentication addresses the security issues inherent in physical access to individual machines or passwords.

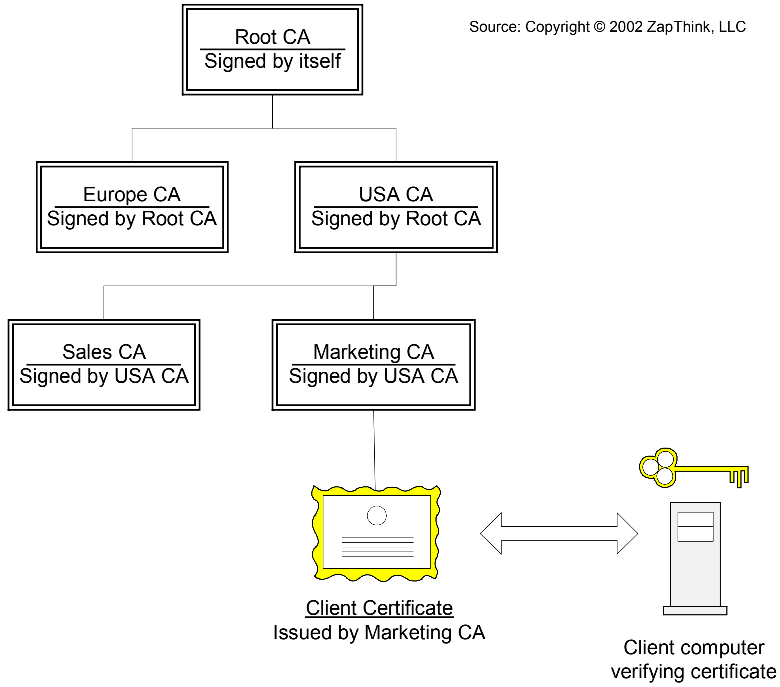
3.3.7 How CA Certificates Establish Trust

Certificate authorities (*CAs*) validate identities and issue certificates. They can be either independent third parties or organizations running their own certificate-issuing server software. Any client or server software that supports certificates maintains one or more trusted CA certificates. These CA certificates determine which other certificates the software can validate—in other words, which certificate issuers the software can trust. In the simplest case, the software can validate only certificates issued by one of the CAs for which it has a certificate. Trusted CA certificates may also be part of a chain of CA certificates, each issued by the CA above it in a certificate hierarchy.

CAs can delegate certificate-issuing responsibilities to subordinate CAs. The ANSI X.509 standard (which is the standard for certificates) includes a model for setting up hierarchies of CAs. In this model, the root CA is at the top of the hierarchy. The root CA signs its own certificate, and then signs the certificates that belong to the CAs that are directly subordinate to the root CA, and the process cascades down the certificate chain. A *certificate chain* is series of certificates issued by successive CAs. Figure 3.6 shows a certificate chain leading from a root CA down through an enterprise's internal CA hierarchy.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Figure 3.6: A certificate chain



To verify that a certificate is valid and trustworthy, the verifying party climbs the certificate chain until it identifies a certificate it can trust. If the verifier trusts the issuer's certificate, the verification is successful. Otherwise, the verifier checks the issuer's certificate to make sure it is properly signed by the next CA up the chain. The verifier then checks that CA's certificate, and repeats the process as many times as necessary until it encounters a CA certificate it can trust.

3.3.8 Managing Certificates

The *Public Key Infrastructure* (PKI) refers to the set of standards and services that facilitate the use of public-key cryptography and X.509 certificates in a networked environment. PKI includes the processes for issuing and storing certificates and keys, renewing and revoking certificates, and procedures for delegating the certificate issuing responsibility.

PKI includes the processes for issuing and storing certificates and keys, renewing and revoking certificates, and procedures for delegating the certificate issuing responsibility.

All certificates are valid for a certain time period. Attempts to use a certificate before or after its validity period will fail. Therefore, CAs must be able to renew certificates. In addition, there is always the possibility that the private key used to create a certificate can become compromised, in which case the CA must be able to revoke the validity of that certificate. CAs typically handle certificate revocation by publishing a *certificate revocation list* (CRL), which is a list of revoked certificates, to a set location, and then checking the list as part of the authentication process.

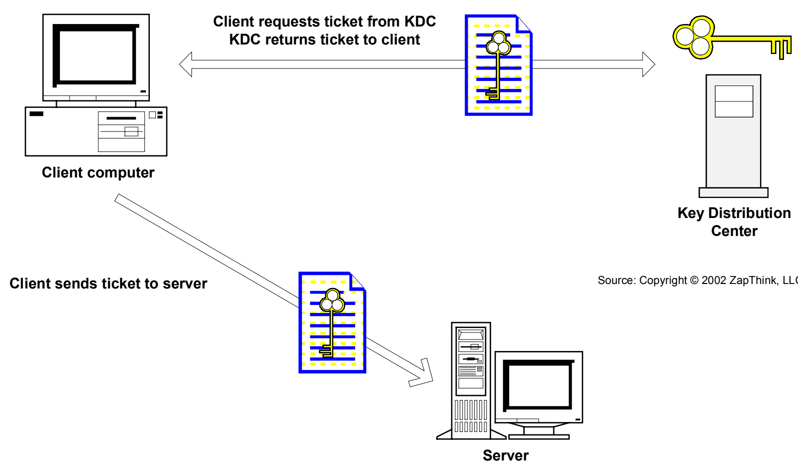
3.3.9 Kerberos

Like public-key encryption, *Kerberos* is an authentication protocol that identifies principals (including users and services) by requiring them to present proof of identity. Kerberos provides for mutual authentication: not only do users of a service identify themselves to the service, but users can challenge the service to prove its identity. To accomplish mutual authentication, Kerberos makes use of *trusted third-party authentication*. Trusted third-party authentication requires that the Kerberos server know who all the parties are as well as how they prove their identity (in particular, their passwords).

The concentration of secret information makes the Kerberos server (technically known as the *Key Distribution Center* or KDC) a very important resource. In particular, companies who use Kerberos must guarantee the physical security of the KDC, and the actual computers housing the KDC may actually be more vulnerable to unauthorized access than the network protocols that shield the KDC.

The Kerberos protocol assumes that all network traffic is vulnerable to capture, examination and substitution. In such an environment, authenticating (or "logging in") to Kerberos is a complex procedure, because the user cannot simply send the password directly to the server, or risk compromising the password. Kerberos solves this problem with encryption technology. The Kerberos client software uses the user's password to generate a symmetric encryption key. After a few exchanges with the server, the KDC returns information to the user, known as a *ticket*, that only software on the workstation that knows the temporary encryption key can use, as shown in figure 3.7. When users wish to contact a Kerberos-protected service, they first contact the Kerberos ticket-granting service and ask for a ticket to the service. A ticket is an encrypted message digest that proves the user's identity to the service.

Figure 3.7: Kerberos authentication



To increase the security Kerberos can provide, it makes use of temporary keys wherever possible. When a user and a service are interacting, they are doing so with a temporary key that the KDC specially generated just for this particular interaction. Such keys expire within a relatively short period of time. Furthermore, Kerberos also applies encryption technology to guarantee data integrity. Assuming that the client and service have authenticated as described above and each have the current temporary key, Kerberos is able to guarantee both confidentiality and data integrity following the digital signature process explained in section 3.3.4.

3.3.10 Basic Security in HTTP

HTTP addresses client authentication in a couple of ways. The HTTP Specification defines an authentication mechanism known as *basic authentication*. In basic authentication, a client passes its credentials (its name and password) to the HTTP service (typically a Web server) in the authentication header of the HTTP request. The HTTP service can authenticate the user or it can pass the credentials to another system for authentication. A second HTTP authentication mechanism, known as *digest authentication*, attempts to strengthen basic authentication by making the client compute a message digest containing the

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

client credentials. Both HTTP approaches have limited value, because neither digest authentication nor basic authentication provides for integrity, confidentiality, or nonrepudiation. For those reasons, browsers use SSL when the user requires more security than HTTP security can provide.

3.3.11 Secure Sockets Layer (SSL)

The *Secure Sockets Layer (SSL)* protocol governs server authentication, client authentication, and encrypted communication between servers and clients. SSL is the most popular security technology used on the Internet, especially for e-Commerce and other interactions that involve exchanging confidential information between a browser and a Web server.

SSL requires a server SSL certificate, at a minimum. As part of the initial handshake process, the server presents its certificate to the client, which the client uses to authenticate the server's identity. The authentication process uses PKI and digital signatures to confirm that the server is in fact the server it claims to be. Once the client has authenticated the server, the client and server use symmetric-key encryption to encrypt all the information they exchange for the remainder of the session, as well as to detect any third-party tampering that may occur during the session.

Servers may also require client authentication as well. In this case, after the client has authenticated the server, the client presents its certificate to the server for it to use to authenticate the client's identity before it establishes the encrypted SSL session.

Decision Point

There are two main limitations to SSL. First, SSL does not provide audit trails, and second, SSL is point-to-point.

There are two main limitations to SSL. First, SSL does not provide audit trails. It is difficult to prove that an SSL session occurred, or even to prove that a recipient received a piece of data sent via SSL. Therefore, users who wish to provide for non-repudiation typically use digital signatures in addition to SSL. The second problem with SSL is that it is point-to-point, meaning that security is guaranteed directly between the two communicating parties. For communications between a browser and a Web server, this limitation is not an issue, but in the more general case where messages can hit several intermediaries between the sender and the recipient, SSL does not provide end-to-end security.

3.4 XML Security Efforts

The principles of application level security, key encryption, digital signatures, PKI, SSL, and Kerberos form the basis for all work in XML security.

The principles of application level security, key encryption, digital signatures, PKI, SSL, and Kerberos form the basis for all work in XML security. XML presents two questions within this existing framework: how should systems secure XML messages, and how can the extensible and open principles of XML apply to the existing security technologies? The two essential XML security efforts that address these issues are XML Signature and XML Encryption.

3.4.1 XML Signature

The *XML Digital Signature (xml-dsig)* standard is a joint initiative of the IETF (Internet Engineering Task Force) and the W3C (World Wide Web Consortium). XML Signature describes a set of XML elements and attributes that store information about the hashing and encryption algorithms that software uses to generate digital signatures, as well as the signatures themselves. In addition, the public key that systems use to verify the digital signature either appears within the XML signature itself, or alternatively the XML signature can include the URL of the public key directory that includes the public key.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

XML Signature brings the well-known benefits of XML to digital signatures, making them human-readable, easily parsed, platform independent, and simpler to implement.

While XML Signature defines how to render digital signatures in XML, it's not the case that XML Signature is solely intended for digitally signing XML documents. In fact, it is for signing any kind of digital content. XML Signature brings the well-known benefits of XML to digital signatures, making them human-readable, easily parsed, platform independent, and generally simpler to implement than earlier digital signature standards. In addition, XML Signature allows for one signature to reference multiple documents, and mandates that software must also sign information about the relevant encryption algorithms.

On the surface, XML Signature seems like a relatively simple extension of the existing work on digital signatures. However, bringing XML into the fray has led to some subtle, but complex issues. For example, the xml-dsig standard mandates that systems should sign only what the user sees. If a system uses a stylesheet to render the XML on a screen, then the server software must sign the visual representation of the data, since this is what the user actually sees. In addition, the xml-dsig standard specifies that the data must be signed along with whatever filters, stylesheets, client profiles or other information that might affect its presentation.

Another tricky problem with XML Signatures is that parties to a message must protect the *document itself*—not just the contents of the XML tags—so that no changes happen to it in transit that could invalidate the signature. The problem with this requirement has to do with the one-way hashing algorithms that systems use to produce the message digests that form a critical part of a digital signature. An XML document typically contains some white space between tags, for example, and this white space may change when a DOM or SAX processes the XML. Furthermore, the order in which tags or attributes occur in an XML document may change when it's loaded into a DOM or SAX processor. Therefore, when the application computes the message digest, then the hash will not match the original hash (because the white space or tag order is different) and so the signature will not be valid. To solve these problems, there are ongoing efforts with specifications known as *XML Canonicalization*. XML Canonicalization defines a standard way to normalize XML information among applications and operating systems. The resulting canonical XML should be application and platform-neutral, so that white space, tag order, and other minor differences caused by different XML parsing techniques disappear.

3.4.2 XML Encryption

XML Encryption is currently a proposal that is working its way through the W3C. In addition to being a means to encrypt XML, XML Encryption also expresses meta-information about signed digital documents, so that processors of those documents are aware of what algorithms were used to encrypt them. XML Encryption also allows for the encryption of entire XML documents or parts of XML documents, allowing systems to use the unencrypted data even when they cannot decrypt the confidential data.

Systems use XML Encryption and XML Signature together when both signing and encrypting a document. In order to check the signature of the data, a system must first decrypt the signed document with a decryption algorithm. This decryption algorithm is also working its way through the W3C. The decryption algorithm proposal notes that when a document is first signed and then encrypted, the XML Signature reveals the message digest value of the signed resource, which is useful information for an attacker.

There are additional problems with XML Encryption. As with general encryption, there's no problem digitally signing an XML document as a whole. However, systems will have difficulty when signing parts of a document, perhaps by

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

XML Encryption is not yet ready for prime-time since many lingering issues remain to be resolved.

different people, especially when each party wants to encrypt their part of the message. Combine these multi-party issues with the XML formatting issues that led to XML Canonicalization, and it is clear that much work remains to resolve the issues behind XML Encryption.

3.5 Web Services Security Efforts

Web Services security efforts fall into three basic categories:

- XML Signature services (xml-dsig), as covered in section 3.3.4
- XML authentication and authorization services (SAML and XACML)
- XML key management services (XKMS)
- WS-Security, which provides a comprehensive Web Services security model that supports and integrates the other efforts to provide broad interoperability

3.5.1 SAML

SAML (the Security Assertion Markup Language) is a standard for exchanging authentication and authorization information between systems or domains. SAML is a reasonably mature specification (for a version 1.0 standard).

The SAML specification includes three types of assertions:

- Authentication assertions (facts that users have proven their identities)
- Attribute assertions (information about the user such as credit limits)
- Authorization decision assertions (whether the user is authorized to buy an item, for example)

SAML concerns itself only with authentication and authorization. Its main goal is to provide a standard procedure for enabling single sign-on across organizational boundaries using Web Services.

SAML concerns itself only with authentication and authorization. Its main goal is to provide a standard procedure for enabling single sign-on across organizational boundaries using Web Services. The SAML protocol describes how systems request and retrieve assertions, using SOAP over HTTP (in SAML 1.0; additional wire protocols should appear in future versions of the specification). The SAML protocol then defines the request and response messages and the simple choreography for using them. SAML allows for 10 subject authentication mechanisms, including user name/password, Kerberos ticket, SSL certificates, PKI public keys, XKMS public keys and XML Signatures.

A client starts a typical SAML exchange by sending a Web Service request containing both a SAML authentication request and the resource the client wants to access. The client making the request can be a person, a server or a Web Service. If the SAML server successfully authenticates the client, it will return a digitally signed security token to the client (using the xml-dsig standard). The token is valid only for a certain amount of time and can restrict the user to particular levels of access, such as read, write or delete. The client can then use the SAML token to make further requests of any system or Web Service that trusts the SAML server, even across company boundaries. This single-sign-on capability allows systems to chain Web Services together, where the same client must authenticate to multiple Services without having to recheck user credentials.

OASIS declared SAML 1.0 final in April 2002, and is expected to ratify it as a standard specification this summer. Vendors such as **Baltimore Technologies**, **Quadrasis**, **Netegrity**, **Systinet**, **Waveset** and **VeriSign** have based products on SAML.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

3.5.2 XACML

XACML (XML Access Control Markup Language), also developed by OASIS, allows systems to express access control policies in XML. XACML is a specification for capturing authorization and entitlement policies for resources. When a Web Service receives a SAML assertion, the Web Service sends a request to a SAML PDP (Policy Decision Point), which then checks an XACML policy via a PRP (Policy Retrieval Point). Using XML for access control allows systems to replicate policies from various access control products easily, using XACML as a common data format. The XACML initiative defines a structured entitlement or policy language, something the SAML committee chose to leave out in its first iteration.

XACML addresses fine-grained control of authorized activities (e.g., read, write, copy, etc.) based on access requester characteristics, the protocol over which the request is made, and the authentication mechanism. It is important to point out that a compliant Policy Decision Point (PDP) may choose a representation entirely different from XACML for its internal evaluation and decision-making processes. That is, XACML may be regarded simply as a policy interchange format, with any given implementation translating the XACML policy to its own local policy language at some point prior to evaluation.

3.5.3 XKMS

XKMS (XML Key Management Protocol), a W3C specification, provides a means for using Web services to simplify a number of complex PKI protocols and processes. XKMS incorporates X-KISS (XML Key Information Service Specification), which provides a means to query the trustworthiness of a user's digital certificate. XKMS also incorporates X-KRSS (XML Key Registration Service Specification), which enables systems to register digital certificates with an XKMS service.

XKMS brings the advantages of XML to PKI. XKMS simplifies the integration of PKI with a wide variety of XML-aware applications. XKMS shifts essential PKI processes away from the desktop to the PKI server.

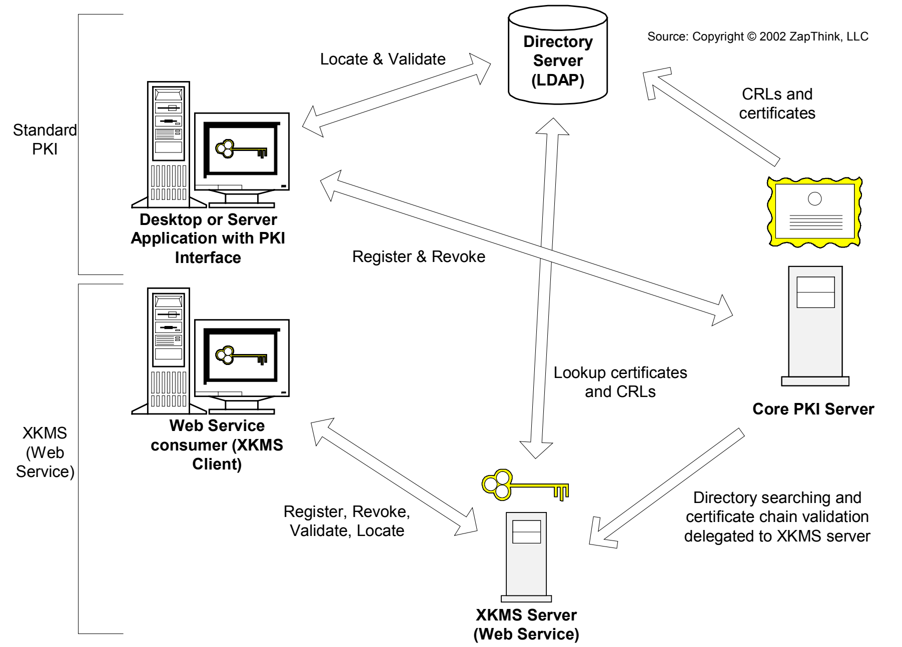
XKMS brings the advantages of XML to PKI. XKMS uses XML to augment PKI in order to outsource key registration, validation, and other trust processes to third party systems. The standard gives developers a common interface for such processes, letting them ignore the underlying PKI. Therefore, developers will be free to focus on Web Services applications rather than the complex tasks associated with PKI.

XKMS simplifies the integration of PKI with a wide variety of XML-aware applications. XKMS deals with the key management aspects of PKI, including the issuing of digital certificates and reporting on their status, but not the cryptographic functions of PKI, namely signing and encryption. XKMS provides a common communication channel for applications that consume PKI services, and shifts essential PKI processes away from the desktop to the PKI server. Companies use cryptographic capabilities in their operating system or existing applications in conjunction with XKMS to support their security requirements. In particular, such companies no longer need to deploy additional PKI toolkits to support multiple PKI protocols.

Because XKMS addresses issues associated with PKI integration, it acts as a front end to PKI to streamline the PKI development effort, as shown in figure 3.8 below. XKMS is an appropriate front end to PKI, because PKI resolves the issues that arise in heterogeneous security environments. In particular, PKI can be invisible to the Web Services implementation, while administrators are still able to use mature PKI products.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Figure 3.8: XKMS as Front End to PKI



✯ Vendor Focus

VeriSign
Microsoft
WebMethods
Baltimore Technologies

VeriSign, **Microsoft**, and **webMethods** created XKMS and submitted it to the W3C. **Baltimore Technologies** also has a leadership role in the development of XKMS. It contains two parts:

- The *XML Key Information Service Specification* (X-KISS) allows a client to delegate to a third-party Trust Service some or all of the tasks necessary to process an XML Signature. This ability is useful for developers who do not want to implement signature checking themselves, or who simply want to delegate this functionality to an Application Service Provider or other third party who might better be able to act as a Trust Service.

- The *XML Key Registration Service Specification* (X-KRSS) is an XML-based replacement for existing binary PKI file formats for when a user applies for a digital certificate.

3.5.4 X-KRSS

X-KRSS supports servers that register, revoke, and recover public and private key pairs. These keys enable a system to authenticate a user, either as part of PKI or another secret key encryption scheme. An X-KRSS registration server acts as an intermediary between the client and the PKI service that stores key and user information. X-KRSS servers serve a similar role to certificate authorities in PKI.

X-KRSS registration supports the following modes:

- The client generates the public and private key pair and sends the public key to the registration server. The server can then ask the client for *proof of possession* (POP) of the matching private key.

- The X-KRSS registration server can store client user names and other attributes that are bound to the client's public key. In this mode, the X-

KRSS server can either generate the certificate or delegate the certificate generation process to another PKI service.

- The X-KRSS server generates the key pair and sends the private key to the client using the XML Encryption. This mode would be useful if the client loses its private key.

3.5.5 X-KISS

The X-KISS protocol provides for assertion servers that can locate and validate keys from an XKMS registration service. By using features in the XML Signature specification, X-KISS can support three tiers of service that describe key location and validation functions that both clients and assertion servers are able to perform:

- Tier 0: The client locates and validates keys.
- Tier 1: The client delegates locating keys to the assertion server, but performs the key validation itself.
- Tier 2: The client delegates both the location and validation of the keys to the assertion server.

Clients operating at tier 0 can support many PKI and digital certificate tasks, as well as XML-related security tasks. For example, if a Web Services client receives a document digitally signed following the XML Signature specification, the client can look inside the signature to determine the URL where it can find the certificate containing the public key. On tier 1, the server locates information describing the public key and sends it to the client, which then validates that information locally. Tier 1 is especially useful in situations where the client doesn't fully trust the server. Finally, tier 2 allows the server to take on both tasks itself, in situations where the client fully trusts the server. Tier 2 is especially suited for clients that do not have the power to perform the key location and validation functions themselves.

3.5.6 WS-Security

Finally, *WS-Security (Web Services Security)* provides a broad set of specifications that cover security technologies that apply to authentication, authorization, privacy and confidentiality, trust, delegation, and auditing (non-repudiation) across a broad range of existing security frameworks and technologies, including both PKI and Kerberos. The WS-Security specification defines a comprehensive Web Service security model that supports and integrates the security models and technologies discussed above at a level of abstraction that enables multiple, heterogeneous systems to interoperate securely. Currently, WS-Security's creators, **IBM**, **Microsoft**, and **VeriSign,** are driving the development of WS-Security. These vendors published the WS-Security 1.0 specification in April 2002, and **IBM** so far is the only company that has released prototype software based on the specification. Microsoft has also announced their TrustBridge initiative which they are basing on WS-Security and Kerberos, but they have not yet released any related software.

☆ Vendor Focus

IBM
Microsoft
VeriSign

WS-Security applies to Service-oriented architectures at a higher level than SAML in that it doesn't establish a universal framework to express user policies, nor does it provide a way to pass security credentials among security domains to build single-sign-on systems. However, WS-Security does support data encryption. SAML, in contrast, currently requires the use of encrypted transport mechanisms like HTTPS (SSL over HTTP) to provide message security.

Because WS-Security works at a level of abstraction above the nuts-and-bolts of application level security, WS-Security does not ensure security by itself, and it

Decision Point

WS-Security does not ensure security by itself, and it doesn't provide a complete security solution. Instead, companies must use WS-Security in combination with other Web Service security protocols.

doesn't provide a complete security solution. Instead, companies must use WS-Security in combination with other Web Service security protocols to accommodate a variety of security models and encryption technologies. WS-Security explicitly builds upon certain foundational technologies, including SOAP, WSDL, XML Signatures, XML Encryption and SSL. The intention is to provide a framework that allows Web Service providers and consumers to develop solutions that meet their own individual security requirements.

The WS-Security specification includes a broad set of specifications including authentication, authorization, privacy, trust, integrity, confidentiality, secure communications channels, federation, delegation and auditing. In addition, WS-Security operates across a broad range of applications and architectures. WS-Security provides an extensible and flexible framework that its creators have especially designed to enable companies to leverage existing investments in security technologies, even when the other parties they wish to securely communicate with use different technologies.

The three vendors who are driving WS-Security are taking a phased approach to rolling out WS-Security. The current effort, which they released in April 2002, consists of a roadmap and an initial "seed" specification. This early draft defines the core facilities for protecting the integrity and confidentiality of messages, as well as mechanisms for associating security-related claims with messages. WS-Security's creators then hope to work with the industry and one or more standards bodies to flesh out the follow-on issues of policy, trust and privacy The roadmap also discusses additional issues, including firewall processing, privacy, browsers and mobile clients, access control, delegation, and auditing.

The initial specifications that make up the current draft of WS-Security include:

- *WS-Security* describes how to attach signature and encryption headers to SOAP messages. It also describes how to attach security tokens to messages, including digital certificates and Kerberos tickets. The term "WS-Security" formally refers to this particular specification as well as referring informally to the overall group of specifications.
- *WS-Policy* describes the capabilities and constraints of both security and general business policies on both message intermediaries and endpoints. For example, WS-Policy describes required security tokens, supported encryption algorithms, and privacy rules.
- *WS-Trust* describes a framework for trust models that enable Web Services to interoperate securely.
- *WS-Privacy* describes a model for how Web Services and requesters can state their own privacy preferences as well as organizational privacy statements.

The follow-on specifications that IBM, Microsoft, and VeriSign are looking to the industry to help develop include:

- *WS-SecureConversation* describes how to manage and authenticate message exchanges between parties, including security context exchanges and establishing and creating session keys.
- *WS-Federation* describes how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities.
- *WS-Authorization* describes how to manage authorization data and policies.

WS-Security takes a much less efficient approach to authentication than SAML, but WS-Security has broader applicability than SAML.

The basic goal of WS-Security is similar to that of SAML, which is to provide interoperable, XML-based mechanisms for providing access to Web Services. Unlike SAML, however, WS-Security doesn't provide for the transfer of security authorization from one system to another, which is necessary to support single sign-on capabilities, because WS-Security doesn't separate the action of authentication from the Web Service request itself. Systems that support WS-Security will either accept or reject the request as a whole, and since every request must include authentication, WS-Security takes a much less efficient approach to authentication than SAML does.

WS-Security also goes further than SAML in providing ways to seal the entire SOAP message against tampering, both by signing it to ensure sender identity and encrypting it to provide confidentiality. SAML provides guarantees of sender identity and confidentiality only for the SAML security tokens themselves, not for the rest of the Web Service request.

WS-Security accomplishes message signing and encryption by using the XML Signature and XML Encryption standards. However, WS-Security does not depend on HTTP to provide these protections, while SAML does, so at this point, WS-Security has broader applicability than SAML. It remains to be seen, however, whether a future version of the SAML specification will either remove this limitation, or move toward a specification that is unified with WS-Security.

IV. Market Segmentation

Some of the vendors in this report may be selling solutions that nobody will ever want to buy.

Making sense out of an emerging market as dynamic as the XML and Web Services security market is especially challenging, for several reasons. Because the demand for many of the solutions available in this space depends on broad implementation and availability of Web Services, many of the companies in this report are basing their strategies on expected future customers. As a result, some may be selling solutions that nobody will ever want to buy. In addition, there is no established terminology for referring to the types of products that are entering this category. Furthermore, most of the XML and Web Services-specific products and services actually overlap existing market segments that fall outside the scope of the XML and Web Services security marketplace.

As a result, one of the goals of this report is to help make sense of this turbulent environment. ZapThink has identified twelve market segments that will roughly group competitors together, as shown in figure 4.1. Note, however, that many vendors appear in more than one segment, and the figure also illustrates overlap among several segments, as well. Both aspects of this market map serve to illustrate the fluid nature of the XML and Web Services security space.

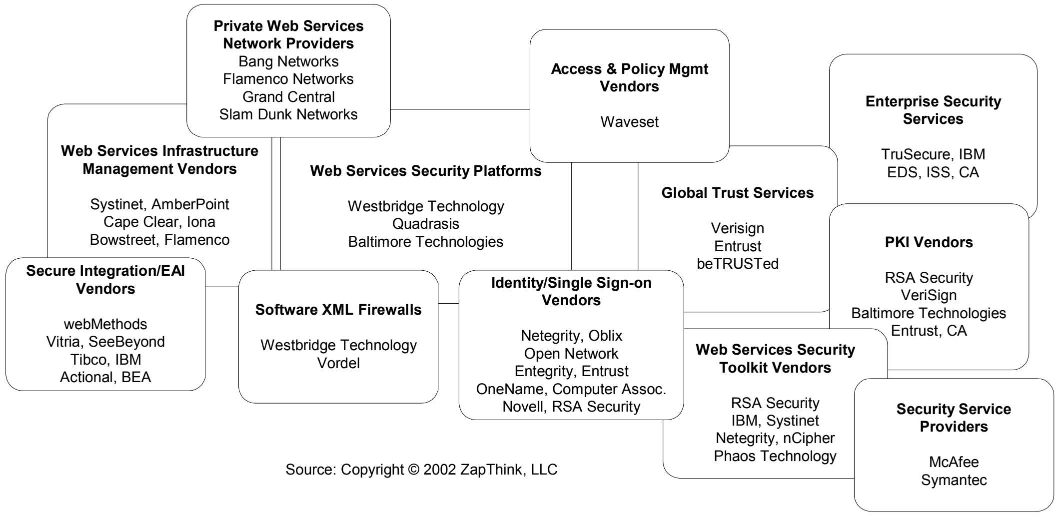
Table 4.1: XML and Web Services Security Markets

Market Segment	Vendors
Web Services Security Platforms	Westbridge Technology Quadrasis Baltimore Technologies
Secure Integration/EAI Vendors	webMethods Vitria SeeBeyond Tibco IBM Actional BEA Systems
Global Trust Services	VeriSign Entrust beTRUSTed
Identity Management/Authorization/Single Sign-On Vendors	Netegrity Oblix OpenNetwork Entegrity Entrust OneName Computer Associates Novell RSA Security
Access and Policy Management Vendors	Waveset
PKI Vendors	RSA Security VeriSign Baltimore Technologies Entrust Computer Associates
Web Services Security Toolkit Vendors	RSA Security IBM Systinet Netegrity nCipher Phaos Technology
Software XML Firewalls	Westbridge Technology Vordel
Private Web Services Network Providers	Bang Networks Flamenco Networks Grand Central Slam Dunk Networks
Enterprise Security Services	TruSecure IBM EDS ISS Computer Associates
Security Service Providers	McAfee Symantec

Source: Copyright © 2002 ZapThink, LLC

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Figure 4.1: The ZapThink XML and Web Services Security Market Map



Source: Copyright © 2002 ZapThink, LLC

This report discusses significant players in each of the segments illustrated above. The companies we consider do not form an exhaustive list; in some cases, we only consider typical companies, not meaning to indicate that missing names are any less important. Section 4 focuses on the market segments, and presents a high-level view of the participants in order to illustrate the types of solutions in the segment. Section 8 below provides a more detailed view of each of the vendors.

4.1 Web Services Security Platforms

The concept of a *platform* is a set of technologies that underlie a variety of applications and services and provide several basic services to those applications and services. It is unusual, therefore, to find a special use platform, because by their very nature, platforms serve multiple purposes. As a result, many existing IT platforms provide certain security features, and many of those also support Web Services, including **IBM** WebSphere, **BEA Systems** WebLogic, as well as the EAI platforms considered below.

There are certain vendors, however, who are developing Web Services platforms that specifically focus on security as the primary value-add of the platform, or at the least, one of a few value-adds. These vendors include **Westbridge Technology**, **Quadrasis**, and **Baltimore Technologies**.

Web Services security platforms typically share many of the following features:

- A security policy engine that coordinates security and privacy policies across multiple systems
- XML network management capabilities, including monitoring and reporting features. Platforms also often integrate with existing network management tools (typically via Web Services interfaces, naturally)
- An XML firewall that inspects SOAP packets and other XML documents and makes decisions based upon their contents (we also consider software XML firewalls as a separate segment in section 4.9)
- A trust server that might manage keys and certificates, and serve as a registration and assertion server
- Adapters and connectors for integration with various other systems
- Many also have support for SAML tokens

Vendor Focus

IBM
BEA Systems
Westbridge Technology
Quadrasis
Baltimore Technologies
Vordel
Actional

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Other vendors who have offerings that overlap the Web Services security platform segment includes software XML firewall vendor **Vordel**. **Actional**, whom we consider to be primarily an integration vendor, offers a policy server as a core part of their solution. In addition, this segment overlaps the Web Services infrastructure management segment, which we consider next.

4.2 Web Services Infrastructure Management Vendors

As we showed in figure 2.1, the management of Web Services is the next roadblock to the adoption of Web Services after security. It is no surprise, therefore, that first, there are many companies competing in this space, and second, many of them incorporate Web Services security as an integral part of their solution.

Vendor Focus

Systinet
AmberPoint
Cape Clear
Iona
Bowstreet
Flamenco Networks

Significant vendors who are providing Web Services infrastructure management solutions that include Web Services security as a core part of their offering include **Systinet**, **AmberPoint**, **Cape Clear**, **Iona**, **Bowstreet**, and **Flamenco Networks**. Each of these vendors typically offers several of the features listed for Web Services security platform vendors as shown in section 4.1 above. In addition, vendors in this segment often offer many of the following:

- Server platforms that support the development and deployment of Web Services
- Web Service management capabilities, including auditing, monitoring, and control capabilities
- Quality of service tools that monitor and manage uptime, availability, and other factors that form service-level agreements (SLAs)
- Web Service provisioning tools and other tools that move Web Service components into production

In addition, vendors like **Systinet** and **Bowstreet** offer development tools as well as platform components.

4.3 Secure Integration/EAI Vendors

Security has been a mainstay of the Enterprise Application Integration (EAI) vendors' solutions since the technology first appeared. Now that every major EAI vendor is providing Web Services interfaces, they are all necessarily in the Web Services security business.

Vendor Focus

webMethods
Vitria
SeeBeyond
Tibco
IBM
BEA Systems
Actional

The four most significant EAI vendors are **webMethods**, **Vitria**, **SeeBeyond**, and **Tibco**. All four vendors are implementing Web Services and offer Web Services security as part of their EAI strategy. In addition to these vendors, **IBM** and **BEA Systems** are continuing to add integration capabilities to their core application server platforms. These vendors also offer Web Services security capabilities. The last vendor we would like to consider in this segment, **Actional**, takes a different approach from the other vendors: instead of EAI or application server middleware, Actional's core product is more of an XML gateway or switch that routes XML traffic instead of acting as a hub or bus, like most EAI systems. Nevertheless, Actional provides a robust integration offering with Web Services and XML security features.

For detailed information on the EAI market, please see our recently released report, *Service-Oriented Integration* (SOI-ZTR-WS103).

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

4.4 Global Trust Services

Vendor Focus

VeriSign
Entrust
beTRUSTed

The advent of PKI required certain companies to be the root CA that can sign certificates for public use, like those in Web servers and browsers. This requirement led companies like **VeriSign** to position themselves as global trust services, offering not only root CA services, but a wide range of application level security services offered via an application service provider (ASP) model. Such vendors, including **Entrust** and **beTRUSTed**, also offer a broad range of software solutions that enterprises can use to establish their own internal or partner trust services.

With the advent of Web Services, as well as the XML security technologies covered in section 3.4, these global trust services have moved into providing their core services for companies requiring XML and Web Services security services delivered over the Internet. Global trust services vendors typically offer:

- Root CA services, including the issuance of digital certificates, certificate revocation list management, and certificate authentication services
- "Private label" PKI services that allow companies to offer root CA services under their own brand
- Outsourced application level security services, including access control provisioning, authentication, and authorization, as well as encryption and digital signing

Many companies implementing single sign-on will also take advantage of global trust services as part of their solution. As a result, single sign-on vendors often integrate with global trust service providers.

4.5 Identity Management/Authorization/Single Sign-On Vendors

This market segment actually comprises three closely related solution areas:

- *Identity management* solutions provide companies with the ability to provision and manage individuals' and other resources' identity information, along with their usernames, passwords, security permissions, and other security and privacy policy information specific to the individuals or resources.
- *Authorization* solutions process user credentials (username, password, client certificate, etc.) in order to create an authorization token, which may or may not be a SAML token.
- *Single sign-on* solutions allow companies to provide their users with a single authorization interface that allows them to access multiple, typically heterogeneous systems, either within a single security domain, or across domains.

Vendor Focus

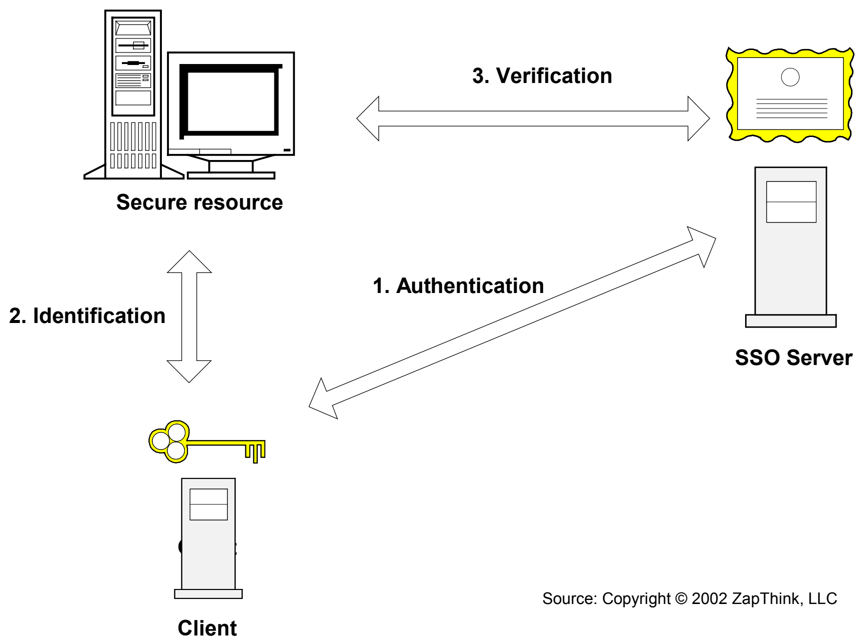
Netegrity
Oblix
OpenNetwork
Entegrity
Computer Associates
Entrust
Novell
RSA Security
OneName
Sun Microsystems

This report combines these three types of solutions, because the vendors who are building products in these categories that also leverage the XML and Web Services security standards discussed in section 3 are all generally providing solutions in all three categories. These vendors include **Netegrity**, **Oblix**, **Open Network**, **Entegrity**, **Computer Associates** (CA), **Entrust**, **Novell**, **RSA Security**, and **OneName**. There are other vendors, like **Sun Microsystems**, who provide identity management solutions as part of their directory server offering. However, the directory server market segment is beyond the scope of this report.

The *single sign-on (SSO)* security architecture shifts the complexity of a heterogeneous security architecture to the SSO service, by allowing individual security mechanisms to delegate the responsibility for identity management and authorization to the SSO server. In the SSO architecture, all security algorithms reside in the single SSO server, which acts as the only authentication point for the domain. As a result, a user need only log in once, even though that user may subsequently interact with many different secure systems within the domain. The SSO server, which can itself be a Web Service, wraps the existing security infrastructure, which in turn exports security features like authentication and authorization to the SSO server.

In a typical SSO scenario, the client first calls the SSO server, provides the correct authentication credentials, and requests the authentication token that identifies it to the desired domain. The SSO server then validates the credentials using the underlying security infrastructure, and only then issues a ticket or other authentication token that the client can use to authenticate in other applications. In this scenario, the token doesn't contain any specific information; it simply identifies the user for some specified scope and time. After the invoked application receives the token, it validates it by sending it to the SSO server that then confirms its validity, as shown in figure 4.2:

Figure 4.2: A Simple SSO Scenario



The advantages of SSO are as follows:

- SSO encapsulates and hides each systems' underlying security infrastructure in the SSO server. Each system in the domain need only interface with the SSO server to support enterprise-wide SSO.
- The SOAP interface to the SSO server makes the SSO architecture universally accessible, because the SSO is itself a Web Service. If the SSO server exposes its interface in a WSDL file, then any Web Service consumer in the domain can access its functionality.
- Because security credentials only go to a single location, the SSO server enhances the security of the whole system. SSO can also provide for

federation across different domains (say, in a multiple company scenario). SSO servers can then authenticate credentials for use outside a particular domain, while the security credentials themselves never leave the domain.

Among the vendors in this market segment, **Netegrity** and **Oblix** are recognized as the leaders in SSO, and both vendors are leaders in the Web Services space, developing solutions that leverage SAML. **Entegrity** is also leveraging SAML, but they target J2EE enterprises, rather than the heterogeneous enterprises that Netegrity and Oblix target. **OpenNetwork**, on the other hand, is a leader in the Microsoft .Net SSO space, and offers Active Directory-based .Net to J2EE SSO solutions. **Entrust** is also a full-service SSO vendor, but their XML and Web Services strategy is still unclear at this time. And finally, **OneName** takes a different approach from the other vendors in this segment, offering a pure Web Services-based identity management solution that can be incorporated into the enterprise with or without an SSO solution.

Vendor Focus

Netegrity
Oblix
Entegrity
OpenNetwork
Entrust
OneName

4.6 Access & Policy Management Vendors

Traditionally, security systems have handled access management with access control lists (ACLs). An ACL will contain a list of users and which resources they are or are not able to access. More sophisticated ACLs also control core database operations: which users can read, modify, or delete which data. Today, however, enterprises are finding that their information access policies are far more complex than a simple ACL can handle. After all, security and privacy policies are responses to business imperatives. Typical policies might include "members of the HR department below the manager level need an HR manager's approval to make purchases over $100." Likewise, a typical privacy policy might be: "Web pages with any personally identifiable information (name, address, employer, associations, etc.) can only be available to employees at the manager level or above, and only when they log in from the internal network."

Clearly, such policies would be very difficult to cast in ACLs. In addition, such policies tend to be quite dynamic, and a company's process for changing policies is itself cast in a set of policies. For these reasons in particular, Web Services are particularly suited for building access and policy management solutions, because they are coarse grained (allowing for the expression of high-level business policies) as well as loosely coupled (allowing for broad flexibility across the enterprise). On the other hand, policy management is a difficult problem, and many solutions will likely take years to mature.

Web Services are particularly suited for building access and policy management solutions, because they are coarse grained and loosely coupled.

Tackling these problems are vendors like **Waveset**, which offers an identity management platform that handles automated provisioning, delegated administration, identity audits and vulnerability detection, as well as allowing users to manage their own identity information. Waveset is helping to drive the work on Web Service provisioning standards in OASIS, and will be supporting SAML tokens. Additional players in this space include **BMC Software** and **Access360**, but neither of these companies has clear XML or Web Services strategies.

Vendor Focus

Waveset
BMC Software
Access360

Vendors in the access and policy management segment typically provide many of the following features:

- Directory server-based identity management services
- User and resource provisioning capabilities, that enable enterprises to enroll and delete users and other resources
- Sophisticated, rule-based policy management engines

- Auditing and intrusion detection capabilities
- SSO management capabilities, often working in close association with one particular SSO product
- Remote hosting services, often in association with a global trust service.

4.7 PKI Vendors

Public key infrastructures are complex, multifaceted solutions, with many elements that must work together, including certificate issuance and storage, key issuance and storage, key and certificate validation, certificate renewal and revocation, and more. In fact, PKI has had limited acceptance simply because of its complexity and high overhead. For the vendors who compete in this space, XML and Web Services offer a set of technologies that can make PKI simpler to set up and manage. In addition, since many XML and Web Services security technologies are PKI-based, the various solutions built upon these technologies must take advantage of existing PKI solutions. XML and Web Services, therefore, provide two kinds of opportunities for vendors in this segment: applying XML and Web Services to PKI solutions, and vice versa.

☆ Vendor Focus

RSA Security
VeriSign
Baltimore Technologies
Computer Associates
Entrust

Each of the five vendors we consider in this segment—**RSA Security**, **VeriSign**, **Baltimore Technologies**, **CA**, and **Entrust**—all appear in other market segments, as well. VeriSign and Entrust both provide global trust services. Baltimore technologies provides a Web Services security platform, and RSA Security provides Web Services security tools. Entrust and CA also provide SSO capabilities. However, what distinguishes these companies from their competitors in each of these segments is their depth of ability in offering PKI solutions.

The XML and Web Services-oriented PKI solutions that vendors in this segment typically offer include:

- XKMS server capabilities, including registration, revocation, validation, and location of certificates, as well as lookup of CRLs
- XKMS, XML Encryption, and XML Signature tools and applications

4.8 Web Services Security Toolkit Vendors

☆ Vendor Focus

RSA Security
IBM
Netegrity
Phaos Technology

Four leading vendors who have developed development tools for XML and Web Services security are **RSA Security**, **IBM**, **Netegrity**, and **Phaos Technology**. Each of these vendors has a depth of experience with PKI, and is leveraging that expertise to provide XML-enabled tools. RSA Security, a leader in encryption technology, offers commercial security development tools as a core part of their business. IBM, on the other hand, has released their XML Security Suite as open source. Netegrity, who is primarily an SSO vendor, is a leader in the SAML initiative, and has released the Netegrity JSAML Toolkit, a Java-based toolkit that provides components for constructing and reading SAML assertions and constructing and interpreting SAML requests and responses. Phaos Technology also offers tools that support SAML as well as XKMS.

The tools that vendors in this segment typically offer include several of the following:

- Tools for implementing SAML
- Tools for implementing XKMS, including the implementation of X-KISS and X-KRSS
- Support for XML Encryption (as well as decryption)
- Support for digital signatures using XML Signature

- Toolkits often include integrated cryptography libraries, which may include XML Canonicalization tools
- A broad range of tools, components, and APIs for PKI and other security infrastructures

☆ Vendor Focus

Westbridge Technology
Vordel

4.9 Software XML Firewalls

A key part of Web Services security platforms like that from **Westbridge Technology** is software that acts as an XML firewall. **Vordel**, in particular, offers a product that specifically serves this purpose. A firewall is a device that acts as a message intermediary, inspecting the traffic that attempts to pass, and either allows or rejects the traffic based on a range of criteria. Traditional hardware firewalls inspect traffic on the packet level, and as such cannot reject traffic based on the structured content (for example, XML-formatted content) it contains. Software XML firewalls are capable of understanding the content that pass through them and appropriately managing that traffic.

There is a relatively new category of hardware device that is sophisticated enough to inspect traffic on the application level, and make policy decisions based upon the content of the messages that attempt to pass through it. Such devices include those provided by **Sarvega** and **Forum Systems** and are detailed in ZapThink's upcoming *XML Proxies* report.

These firewalls typically operated by inspecting SOAP message headers. If a header has instructions for the XML firewall (which may or may not be signed), the software can make routing decisions based upon those instructions. Even for the part of the SOAP message not intended for the firewall itself, the software firewall may be able to decrypt part or the entire message and make routing or other policy decisions based upon the secure contents of the message.

XML firewalls typically offer several of the following features:

- SOAP message and other XML message inspection
- Malicious attack protection and intrusion detection
- Authentication and authentication capabilities
- Decryption and encryption capabilities
- Non-repudiation capability
- XML Schema and XPath-based rule enforcement
- Real-time monitoring and reporting, typically via a Web Services interface

Web Services offer great potential for B2B communication and integration, but the lack of robust security and manageability solutions currently inhibit the ability for companies to conduct business with each other via Web Services over the Internet.

4.10 Private Web Services Network Providers

In the early days of B2B eCommerce—even before people called it that—companies would use EDI over private networks called *Value-Added Networks*, or VANs. VANs tended to be expensive, but offered reliable wide-area connectivity in the days before the Internet was a commercial tool. Today, Web Services offer great potential for B2B communication and integration, but the lack of robust security and manageability solutions currently inhibit the ability for companies to conduct business with each other via Web Services over the Internet.

As a result, there is opportunity for a new generation of VAN, what might be called a "Web Services VAN," which operates as a private Web Services network. Leveraging Internet technologies, a Web Services VAN can connect a company to its business partners by providing a robust security infrastructure on top of a private network.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

☆ Vendor Focus

Grand Central
Bang Networks
Slam Dunk Networks
Flamenco Networks

Grand Central is one of the emerging leaders in this new market segment. Grand Central provides a private network offered as a subscription service, where companies and their partners can access the network via Web Services and other Internet-friendly protocols. Grand Central's network then offers security, messaging, orchestration, message transformation, and monitoring. Other emerging vendors in this sector include **Bang Networks** and **Slam Dunk Networks**.

Another player in this space is Flamenco Networks. However, **Flamenco Networks** is more of a Web Services infrastructure management vendor who optionally offers its solution via a hosted ASP model. Grand Central, on the other hand, only offers its Web Services VAN as a service.

The reason that private Web Services network providers fall into the broader XML and Web Services security market is because security is the core of their service. Private Web Services network providers typically offer many of the following security features:

- Multiple levels of authentication, typically offered in association with a global trust network
- Encrypted messaging
- Complete access control
- PKI proxy services for storing PKI keys, certificates and relevant information
- Reliable, guaranteed message delivery
- Message transformation services
- End-user "on-ramping" or "provisioning".

4.11 Enterprise Security Services

☆ Vendor Focus

IBM
EDS
Internet Security Systems
Computer Associates
TruSecure

ZapThink believes that it is important to discuss enterprise managed security services in this report, not because there are many service providers in this segment who are offering XML or Web Services-related managed security services today, but rather because Web Services clearly provide a growth area for these providers. There are many significant vendors in this space; a representative sample includes **IBM**, **EDS**, **Internet Security Systems** (**ISS**), **CA**, and **TruSecure**.

These service providers typically take a "just-in-time" proactive approach to emerging technologies. On the one hand, if a new technology like Web Services does not yet present a risk to their customers, then they need not provide managed Web Services security services. However, once their customers begin to ask about Web Services security, or more significantly, begin Web Services pilot projects that present possibly misunderstood risks to the enterprise, then it is the role of the managed security service provider to proactively step in and provide the necessary information and services to insure their customers remain secure.

Therefore, none of the four vendors listed above has an explicit Web Services strategy, but all are necessarily aware of the potential risks, and will be able to provide managed security services for Web Services when their customers demand it.

4.12 Security Service Providers

☆ Vendor Focus

McAfee
Network Associates
Symantec

The last segment this report will consider contains vendors who provide security functionality via a service-oriented model. In particular, **McAfee** (a division of

Network Associates) and **Symantec** are the leaders in the anti-virus software market, and both vendors offer anti-virus protection as a Web-based service.

Note, however, that we say "Web-based service" instead of "Web Service," because at this time, neither vendor offers their anti-virus service via the XML-based open protocols that characterize Web Services. Nevertheless, both vendors are leaders in the Web-based security service arena, and both vendors are also considering a move to the XML-based Web Services protocols that will characterize their services as Web Services.

V. Current State of the Market

As Web Services are an emerging market, and security is at the leading edge of that market, the XML and Web Services security market is necessarily quite turbulent. Most of the development in this space is taking place in advance of predicted demand for the resulting products and services. However, it remains unclear the level and timeframe for that demand to materialize. Vendors are jostling for solutions that offer barriers to entry; some use patent protection to provide this competitive advantage, but more often than not, the barrier to entry consists of the level of sophistication, and hence number of required person-hours, of the solutions. As a result, some of the solutions that are appearing are quite sophisticated and broadly applicable to customer scenarios that are for the most part still on the drawing board.

5.1 Approaches to the Market

ZapThink sees three general approaches to the market that companies are taking, depending upon the nature of their organizations. These approaches are the following:

- Focused startups that have managed to navigate the very difficult current venture capital climate, and therefore have reasonably sound business models, solid funding, and outstanding engineering teams.
- Somewhat established Web Services vendors, who for the most part are typically still privately held, and are adding security features and functionality to their existing Web Services product lines.
- Larger, typically public vendors who have established businesses outside the XML and Web Services arena, who see this new segment as providing growth opportunities for them. Such vendors are either coming from the IT security space or the distributed computing space.

5.1.1 Focused technology startups

In contrast to the "dot.com" era, the early XML and Web Services security market through 2002 has been characterized by limited, wary venture capital funding. While there is still investment money to be had, many companies must have solid business plans and outstanding executive teams to have any chance at funding. Therefore, companies that have received funding—early rounds in particular—during this period are likely to have both critical success factors. Two notable companies that have successfully run this Venture Capital gauntlet include **Westbridge Technology** and **AmberPoint**.

Seasoned engineering talent is also more available now than it was two years ago, and as a result these companies are able to staff their technology teams with experienced veterans. Some companies like **Systinet** leverage high quality offshore engineering teams. The combination of adequate funding, solid business models, seasoned management teams, and high quality engineering

☆ Vendor Focus

Westbridge Technology
AmberPoint
Systinet

The combination of adequate funding, solid business models, seasoned management teams, and high quality engineering staff leads startups to offer surprisingly robust XML and Web Services security solutions.

staff leads the relatively small number of companies in this category to offer surprisingly robust XML and Web Services security solutions to the marketplace.

- Strengths: Funding, management teams, business models, and engineering staff, all leading to agility in the marketplace
- Weaknesses: Narrow focus, lack of clear demand, vulnerability to the turbulent marketplace
- Opportunities: Establish a solution that provides sufficient barriers to entry
- Threats: Turbulence of market, marketing power and customer relationships of established vendors

5.1.2 "Established" Web Services vendors

Vendor Focus

Cape Clear
Grand Central
Actional

Even though Web Services consist of emerging technologies, there are some companies that have relatively (albeit weakly) established Web Service products and are now broadening their security offerings. However, many of these companies are still young startups with an unclear demand for their products or services. Notable companies in this category include **Cape Clear**, **Grand Central**, and **Actional**.

These companies leverage existing technology and an existing customer base to make rapid inroads into the XML and Web Services security market. However, while security tends to be an integral part of each of these companies' product lines, and they are unlikely to define themselves as security companies. Nevertheless, they will increasingly find themselves competing with companies who do define themselves as security companies.

- Strengths: Existing customer base, core technology offerings, understanding of Web Services marketplace
- Weaknesses: Lack of clear demand, vulnerability to the turbulent marketplace
- Opportunities: Leverage existing technology to offer broad solution to customers
- Threats: Turbulence of market, marketing power and customer relationships of larger established vendors

5.1.3 Larger public vendors

Vendor Focus

VeriSign
Entrust
RSA Security
OpenNetwork
Netegrity
Baltimore Technologies
IBM
WebMethods
Tibco
Iona
Quadrasis
Hitachi

Large, public vendors who compete in the XML and Web Services Security space fall into two broad divisions: established security companies like **VeriSign**, **Entrust**, **RSA Security**, **Open Network**, **Netegrity**, and **Baltimore Technologies**, and established distributed computing companies, including **IBM**, **webMethods**, **Tibco**, and **Iona**. **Quadrasis**, which is a division of **Hitachi**, also falls in this category. In any case, Web Services present clear opportunities for growth for these companies, and all of them, regardless of whether they define themselves as security companies, have deep expertise in application level security.

Many of these vendors take a "suite" approach, offering their customers a broad spectrum of IT products and services that work well together. Web Services, then, are typically an additional interface technology for their suites, and security appears throughout their offerings. Established companies compete on existing customer relationships and mature, broad technology solutions.

- Strengths: Existing customer base, deep technology expertise, financial resources, name recognition
- Weaknesses: Lack of agility, need to keep existing customers happy, ongoing weakness in IT marketplace

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

- Opportunities: New markets to enter, and in some cases, new companies to acquire
- Threats: Turbulence of market, financial risk of entering emerging market, agility of smaller players

5.2 Customer perspective

There are two important questions this report addresses in this section: why do companies invest in IT security, and why will they invest in XML and Web Services security? Both questions are more complex than they may seem on first glance, because of the nature of security. Security solutions address risk, and risk by its very nature implies uncertainty. There are some fundamental principles of risk and security that all companies must follow:

- There is no such thing as perfect security. Perfectly secure networks and systems are an impossibility. The most secure system is one that is turned off, and even that can be compromised.
- It is likewise impossible to write a list of all the risks presenting a company. No matter how complete an attempt at such a list might be, there is always the possibility of an entirely unexpected risk.
- Security is never complete. There will always be new vulnerabilities and new modes of attack. Security is an ongoing, ever-changing process.
- There is no way to accurately calculate the risks that security addresses. Any security failure may lead to unexpected costs and other consequences. Therefore, there is no way to accurately calculate the ROI of a security solution.
- Broad estimates of business losses due to security breaches do not apply to individual companies. Any company that believes it can proactively calculate its losses due to lack of security is simply fooling itself.

There is no way to accurately calculate the ROI of a security solution.

One conclusion to be drawn from the principles above is that any approach to security must be heuristic, or in other words, "seat of the pants," at some level. How much security is really enough? There is no truly objective answer to this question. Each executive must make their own decisions for their companies based upon their resources, personal experience and intuition.

Decision Point

There is a market for XML-based security solutions for companies looking to address current risks unrelated to Web Services.

How, then, will customers approach the XML and Web Services security market? These solutions address either current or future risks. When executives consider the current risks in the enterprise, they will look for the most cost-effective solutions that afford the subjective level of security that they wish to achieve. In some cases, XML-based security technologies will be a cost effective option for existing security infrastructures, like PKI. Therefore, there is a market for XML security solutions for companies looking to address current risks unrelated to Web Services. However, this part of the market is quite small in relation to the future potential of the XML and Web Services security market.

Decision Point

Companies planning on using Web Services across the firewall will necessarily have to resolve the resulting security issues first.

For companies building Web Services solutions, there is a clear transition point where security becomes a critical issue: when they begin to access and/or provide Web Services outside the enterprise firewall—in other words, when they begin projects where they do not control both endpoints of every SOAP message. The lack of adequate security solutions provides a roadblock to companies looking to implement such solutions, as this report discussed previously. Companies planning on using Web Services across the firewall will necessarily have to resolve the resulting security issues first.

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Therefore, the current state of the market consists of a pent-up demand caught in a Catch-22. B2B Web Services projects must wait until sufficient security solutions are available, but there won't be a clear demand for those services until enough companies tackle B2B Web Services projects. As usual, the way out of this vicious circle will be found by the early adopter companies, who see a high enough probability of getting a jump on their competition by investing in as yet unproven Web Services security products.

VI. Business & Technology Trends

ZapThink sees that in late 2002 through 2003, early adopter companies will begin to implement secure B2B Web Services solutions with a small number of willing partners. These projects will form a testing ground for many of the products in the Web Services security space, and will allow vendors to improve their products as well as their strategies. In addition, ZapThink predicts that many vendors will find their products or strategies lacking in this early adopter environment, and will have to change their approach or risk company failure.

There will be a spike in demand for Web Services security solutions within the next 12 months.

Furthermore, the fact that companies must address security before implementing risky projects, combined with the "20 doors" problem in which all security holes must be closed before a domain is secure means that there will be a spike in demand for broad, robust Web Services security solutions within the next 12 months. Just as there is a solid ROI story this year for companies looking to reduce the cost of internal integration with Web Services technologies, ZapThink sees a solid ROI story 12 to 18 months out for companies looking to conduct B2B communications with selected partners.

Web Services will not play a major role in transactional environments in 2002-2003.

The 2003 timeframe won't see many multiple-company B2B Web Services, because companies will choose to implement B2B Web Services on a point-to-point basis.

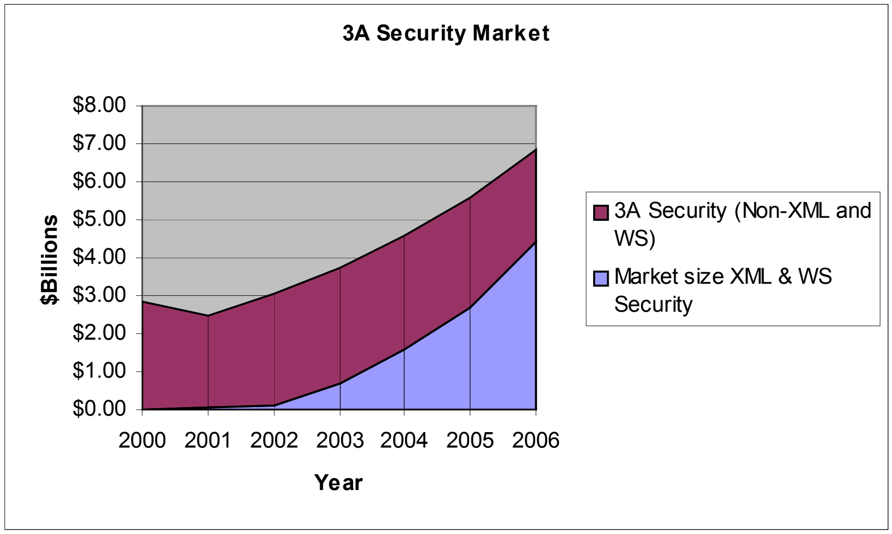
In the 2003 timeframe, however, we do not believe that Web Services will play a major role in true transactional environments, which are those business relationships that must follow the strict ACID (atomicity, consistency, isolation and durability) rules for transactions, because Web Services transactions are a roadblock we believe won't be sufficiently surpassed until later into 2004. Furthermore, the 2003 timeframe won't see many multiple-company B2B Web Services implementations (for example, eMarketplaces), because those companies who are willing to experiment with B2B Web Services will choose to do so on a point-to-point basis, where they fix the Web Services interfaces at design time.

6.1 Long Term Trends: Relationship to the 3A Security Market

Recall from earlier in this report that we consider the XML and Web Services security market to be largely a subset of the application level security, or authentication, authorization, and administration (3A) security market. This characterization excludes network appliance hardware as well as XML and Web Services-related services.

Analysts who cover the entire IT security market estimate the 3A security market to be about $2.5 billion in 2001, rising at a CAGR (compound annual growth rate) of approximately 22% to approximately $6.8 billion by 2006, as shown in figure 6.1 and table 6.1.

Figure 6.1: The XML and Web Services Security Market vs. 3A Security Market



Source: Copyright © 2002 ZapThink, LLC

Table 6.1: The XML and Web Services Security Market vs. 3A Security Market

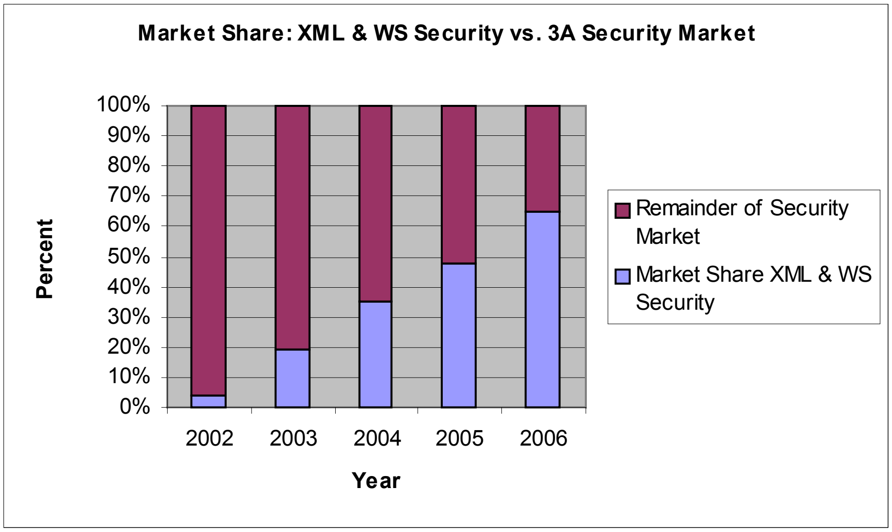
	2000	2001	2002	2003	2004	2005	2006
Market size XML & WS Security	$0.01	$0.04	$0.12	$0.71	$1.60	$2.68	$4.44
3A Security (Non-XML and WS)	$2.82	$2.47	$2.94	$3.03	$2.97	$2.90	$2.39
Total 3A Security Market Size	$2.83	$2.51	$3.06	$3.74	$4.57	$5.58	$6.83

$Billions Source: Copyright © 2002 ZapThink, LLC

The XML and Web Services security market will reach $4.4 billion in 2006, which will represent 65% of the total 3A security market. This growth represents an average CAGR of 300%.

ZapThink believes that the XML and Web Services security market will begin its growth pattern this year (2002), and by 2006, reach $4.4 billion. This market will represent 65% of the total 3A security market by 2006, as shown in figure 6.2. This growth represents an average CAGR of approximately 300%.

Figure 6.2: The XML and Web Services Security Market Share of total 3A Security Market



Source: Copyright © 2002 ZapThink, LLC

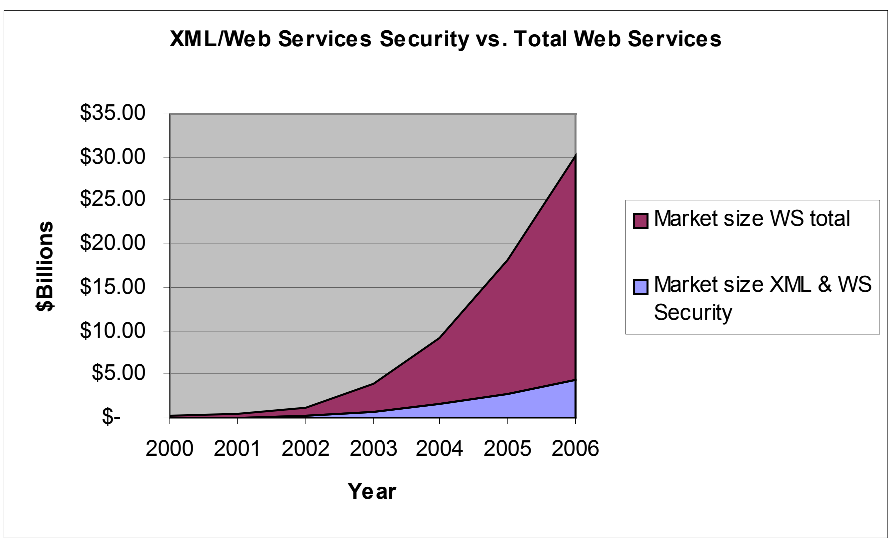
Existing 3A security vendors will incorporate XML and Web Services into their product lines, so by 2006, most 3A security products will support or provide XML and/or Web Services security.

The above graphs reflect two critical forces at work: first, the market for XML and Web Services security products will grow of its own accord, that is, solely as a result of increased opportunities in the XML and Web Services space. At the same time, existing 3A security vendors will increasingly incorporate XML and Web Services into their product lines, so by 2006, most 3A security products will support or provide XML and/or Web Services security.

6.2 Long term trends: relationship to Web Services market

While the XML and Web Services security market will subsume almost two thirds of the overall 3A security market, this segment will form a relatively consistent portion of the total Web Services market, growing at about the same rate as the overall Web Services market, as shown in figure 6.3 and table 6.2.

Figure 6.3: The XML and Web Services Security Market vs. Web Services Market



Source: Copyright © 2002 ZapThink, LLC

Table 6.2: The XML and Web Services Security Market vs. Web Services Market

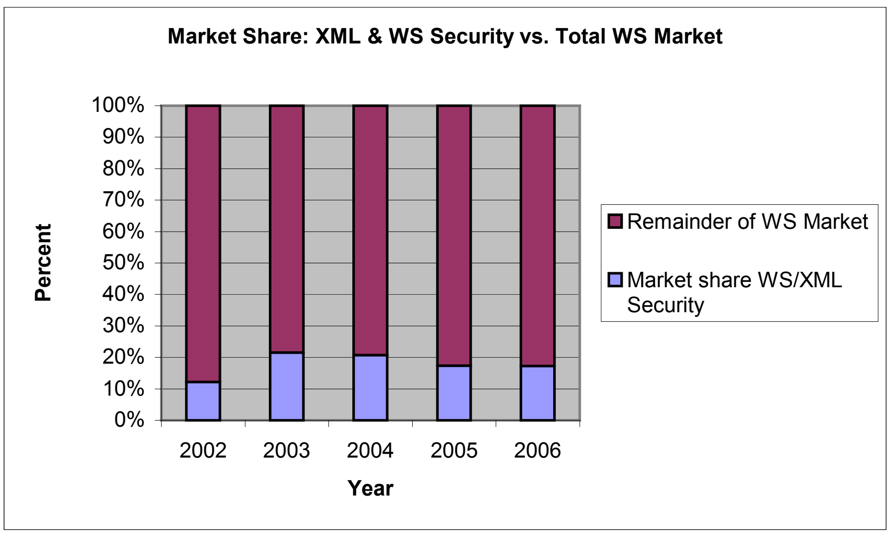
	2002	2003	2004	2005	2006
Market size XML & WS Security	$0.12	$0.71	$1.60	$2.68	$4.44
Remainder of WS market	$0.86	$2.59	$6.10	$12.72	$21.28
Market size WS total	$0.98	$3.30	$7.70	$15.40	$25.72

$Billions

Source: Copyright © 2002 ZapThink, LLC

ZapThink believes that the XML and Web Services security segment will grow most rapidly with respect to the overall Web Services marketplace in the 2003 timeframe, for two reasons: first, security is the next roadblock impeding Web Services implementations, so there will be some pent up demand to satisfy, and second, companies who are looking for security solutions are more likely to purchase a broad solution rather than a limited one (the 20 doors issue again). However, as Web Services take hold in the enterprise over the 2004-2005 time period, security will settle down to being less than 20% of the overall Web Services market, as shown in figure 6.4.

Figure 6.4: The XML and Web Services Security Market Share of total Web Services Market



Source: Copyright © 2002 ZapThink, LLC

6.3 Inhibitors to the Growth of the XML and Web Services Security Market

Enterprises that are tackling their security issues must address both the perception as well as the reality of the risks facing the corporation. In some ways, the *perception* of risk is every bit as dangerous as real risk: it can generate bad publicity, send stock prices down, and chase away customers. To address these issues of perception, executives must make choices that their critical audiences *perceive* as providing security—whether or not they actually do. Because the Web Services market is still emerging, security products in this space will be saddled with the perception that they are new, untested, and less robust than their more established counterparts in the tightly coupled IT world. Regardless of how justified this perception is, it will be an inhibitor to the growth of this market segment.

On the technology side, the largest inhibitor ZapThink sees is the problem of XML encryption. The work on XML Canonicalization is progressing, but nevertheless indicates that handling the encryption and signing of XML messages is problematic. Furthermore, resolving the issues about how to encrypt various parts of SOAP messages is also a complex, difficult issue. We feel that vendors still have a large amount of work left to do before the signing and encrypting of SOAP messages is a generally understood practice.

There are also many issues to be resolved with the specifications that underlie XML and Web Services security. While SAML tokens are supposed to work with the WS-Security specification, there is still no broad consensus in the industry as to exactly how this combination should work. Furthermore, the growing division between Microsoft's Passport and TrustBridge on the one hand, and Sun and the Liberty Alliance on the other could delay the broad acceptance of interoperable approaches to global security.

VII. Conclusions

Security is the next major roadblock impeding Web Services adoption. Therefore, many vendors are focusing their resources to develop solutions to the multitude of security-related problems that companies will face as they seek to implement XML and Web Services solutions across the enterprise, and in particular, among business partners.

Nevertheless, XML and Web Services security is on the front edge of an emerging market, and as such, the entire segment is facing a period of dramatic turbulence. Business models will continue to shift, competitors will enter and leave the market, and companies will face mergers, acquisitions, and failures. Much of this turbulence will occur in a period before there is solid, well-understood demand for the products and services themselves.

The best positioned companies to be profitable in the XML and Web Services security space are those companies that already have deep technical knowledge of application level security technologies, coupled with a solid customer base.

The best positioned companies to be profitable in the XML and Web Services security space are those companies that already have deep technical knowledge of application level security technologies, coupled with a solid customer base. As those customers demand Web Services security, the established security vendors must be able to rise to the occasion.

However, we are not saying that there aren't opportunities for startups and other new entrants to the XML and Web Services security marketplace. As Web Services move beyond the initial "horseless carriage" phase, and enterprises realize that they can leverage open standards-based, loosely coupled Web Services to transform the way they use IT, there will continue to be opportunities for smaller, nimble players who have the resources and courage to push the envelope.

7.1 Key Notes

- *Locking 19 entrances to a building does not provide security if there are 20 entrances.*
- *The primary use for Web Services today is for internal integration.*
- *The next roadblock on the path to Web Services adoption is security. Security is today's key enabler for Web Services.*
- *Most XML and Web Services security offerings on the market fall within the 3A (authentication, authorization, and administration) segment of the IT security market.*
- *Most 3A products will be Web Service-enabled by 2005.*

- *XrML-based DRM solutions will not form an independent market by themselves. Instead, the DRM space will eventually merge with the work in XML and Web Services security.*
- *Application level security focuses on which services users can request, based upon what roles those users have in the system and also provides for the confidentiality and integrity of the transmitted data.*
- *There must be an identification mechanism outside of the software system that provides the initial confirmation of identity.*
- *The ability to provide and administer authorization across multiple systems is a difficult problem that many of the companies covered in this report are trying to solve.*
- *Intermediaries are especially important in the context of Web Services, because the SOAP protocol is designed to support intermediaries that can forward or reroute SOAP messages.*
- *Certificate-based authentication is more secure than simple password-based authentication, because the user must have the private key as well as the password.*
- *PKI includes the processes for issuing and storing certificates and keys, renewing and revoking certificates, and procedures for delegating the certificate issuing responsibility.*
- *The principles of application level security, key encryption, digital signatures, PKI, SSL, and Kerberos form the basis for all work in XML security.*
- *XML Signature brings the well-known benefits of XML to digital signatures, making them human-readable, easily parsed, platform independent, and simpler to implement.*
- *XML Encryption is not yet ready for prime-time.*
- *SAML concerns itself only with authentication and authorization. Its main goal is to provide a standard procedure for enabling single sign-on across organizational boundaries using Web Services.*
- *XKMS brings the advantages of XML to PKI. XKMS simplifies the integration of PKI with a wide variety of XML-aware applications. XKMS shifts essential PKI processes away from the desktop to the PKI server.*
- *WS-Security takes a much less efficient approach to authentication than SAML, but WS-Security has broader applicability than SAML.*
- *Some of the vendors in this report may be selling solutions that nobody will ever want to buy.*
- *Web Services are particularly suited for building access and policy management solutions, because they are coarse grained and loosely coupled.*
- *Web Services offer great potential for B2B communication and integration, but the lack of robust security and manageability solutions currently inhibit the ability for companies to conduct business with each other via Web Services over the Internet.*
- *The solutions that are appearing are quite sophisticated and broadly applicable to customer scenarios that are for the most part still on the drawing board.*
- *The combination of adequate funding, solid business models, seasoned management teams, and high quality engineering staff leads startups to offer surprisingly robust XML and Web Services security solutions.*
- *There is a market for XML security solutions for companies looking to address current risks unrelated to Web Services.*
- *There will be a spike in demand for Web Services security solutions within the next 12 months.*
- *Web Services will not play a major role in transactional environments in 2002-2003.*

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

- *The 2003 timeframe won't see many multiple-company B2B Web Services, because companies will choose to implement B2B Web Services on a point-to-point basis.*
- *The XML and Web Services security market will reach $4.4 billion in 2006, which will represent 65% of the total 3A security market. This growth represents an average CAGR of 300%.*
- *Existing 3A security vendors will incorporate XML and Web Services into their product lines, so by 2006, most 3A security products will support or provide XML and/or Web Services security.*
- *The best positioned companies to be profitable in the XML and Web Services security space are those companies that already have deep technical knowledge of application level security technologies, coupled with a solid customer base.*

7.2 Decision Points

- *This report must be placed into the context of an overall security strategy. Simply securing all of a company's Web Services alone only provides a false sense of security.*
- *Next-generation firewalls must be capable of looking at the content of XML streams, and the security mechanisms for such data must be part of that content.*
- *Enterprises must institute policies that apply to their entire enterprise network (including participants invited from outside), and administer that security in a hierarchical fashion.*
- *There are two main limitations to SSL. First, SSL does not provide audit trails, and second, SSL is point-to-point.*
- *WS-Security does not ensure security by itself, and it doesn't provide a complete security solution. Instead, companies must use WS-Security in combination with other Web Service security protocols.*
- *There is no way to accurately calculate the ROI of a security solution.*
- *Companies planning on using Web Services across the firewall will necessarily have to resolve the resulting security issues first.*

7.3 Figures

7.4 Tables

VIII. Vendor Profiles

8.1 Web Services Security Platforms

Westbridge Technology

Please see ZapNote ZTZN-0612

Quadrasis

Please see ZapNote ZTZN-0304

Baltimore Technologies

Please see ZapNote ZTZN-1008

8.2 Secure Integration Vendors

Actional

Please see ZapNote ZTZN-0280

8.3 Global Trust Services

VeriSign

Please see ZapNote ZTZN-1102

Entrust

Please see ZapNote ZTZN-1038

8.4 Identity Management/Authorization/Single Sign-On Vendors

Netegrity

Please see ZapNote ZTZN-1069

Oblix

Please see ZapNote ZTZN-1073

OpenNetwork

Please see ZapNote ZTZN-1075

Entegrity

Please see ZapNote ZTZN-1037

OneName

Please see ZapNote ZTZN-1074

8.5 Access & Policy Management Vendors

Waveset

Please see ZapNote ZTZN-1106

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

8.6 Software XML Firewalls

Vordel

Please see ZapNote ZTZN-0238

8.7 PKI Vendors

RSA Security

Please see ZapNote ZTZN-0610

8.8 Enterprise Security Services

TruSecure

Please see ZapNote ZTZN-0613

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

A. Related Research

Reports

- *Web Services Technologies and Trends* Report (ZT-WEBSRV)
- *Pros and Cons of Web Services* Report (ZTR-WS101)
- *Service-Oriented Integration* Report (ZTR-WS103)
- *XML Proxies* Report (ZTR-DI101)
- *Service-Oriented Management* Report (ZTR-WS106)

ZapNotes

- *BEA* ZapNote (ZTZN-1009)
- *ContentGuard* ZapNote (ZTZN-0213)
- *Forum Systems* ZapNote (ZTZN-0212)
- *Grand Central* ZapNote (ZTZN-0215)
- *IBM* ZapNote (ZTZN-1050)
- *InterTrust* ZapNote (ZTZN-0265)
- *IONA* ZapNote (ZTZN-0140)
- *Microsoft* ZapNote (ZTZN-1066)
- *Novell* ZapNote (ZTZN-1071)
- *Sarvega* ZapNote (ZTZN-0271)
- *SeeBeyond* ZapNote (ZTZN-0279)
- *Sun Microsystems* ZapNote (ZTZN-1093)
- *Systinet* ZapNote (ZTZN-1096)
- *TIBCO* ZapNote (ZTZN-1100)
- *webMethods* ZapNote (ZTZN-1107)

B. Supporting Resources

XKMS www.w3.org/TR/xkms
XML Signatures www.ietf.org/html.charters/xmldsig-charter.html
XML Encryption www.w3.org/Encryption
SAML www.oasis-open.org/committees/security/index.shtml
W3C www.w3.org
OASIS www.oasis-open.org
IBM's XML Security Suite www.alphaworks.ibm.com/tech/xmlsecuritysuite

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

Trademark Notice and Statement of Opinion

About ZapThink, LLC

ZapThink is an IT market intelligence firm that provides trusted advice and critical insight into XML, Web Services, and Service Orientation. We provide our target audience of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

ZapThink's role is to help companies understand these IT products and services in the context of SOAs and the vision of Service Orientation. ZapThink provides market intelligence to IT vendors who offer XML and Web Services-based products to help them understand their competitive landscape and how to communicate their value proposition to their customers within the context of Service Orientation, and lay out their product roadmaps for the coming wave of Service Orientation. ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into how to assemble the available products and services into a coherent roadmap to Service Orientation. Finally, ZapThink provides demand intelligence to IT vendors and service providers who must understand the needs of IT users as they follow the roadmap to Service Orientation.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOAs by vendors, end-users, and the press. They are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry.

ZapThink was founded in October 2000 and is headquartered in Waltham, Massachusetts. Its customers include Global 1000 firms, public sector organizations around the world, and many emerging businesses. ZapThink Analysts have years of experience in IT as well as research and analysis. Its analysts have previously been with such firms as IDC and ChannelWave, and have sat on the working group committees for standards bodies such as RosettaNet, UDDI, CPExchange, ebXML, EIDX, and CompTIA.

Call, email, or visit the ZapThink Web site to learn more about how ZapThink can help you to better understand how XML and Web Services impact your business or organization.

ZAPTHINK CONTACT:
ZapThink, LLC
11 Willow Street
Suite 200
Waltham, MA 02453
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com