



BRINGING BALANCE TO WEB SERVICES

SOLVING THE TRUST & THREAT EQUATION

The 1980's ushered in the Internet-age, allowing us to connect systems together using a common communication fabric – TCP/IP (Transport Control Protocol/Internet Protocol). The 1990's gave us HTTP (Hyper Text Transport Protocol), and the World Wide Web, for ubiquitous information access using a standard browser. This decade is witness to another significant advance in the Internet as it evolves from the “infrastructure” phase to the dynamic delivery of information-based Web Services. Usage of Web Services ranges from raw computational power, to transactional and multi-media applications for both consumers and business. Today, it is possible to “plug” a wide array of devices into the Internet – including mobile phones, business applications, consumer devices, etc. – as collaborative automation accomplishes tasks without human intervention.

To accomplish this feat, machine-machine interaction takes center stage as machines learn intelligent communication skills using Web Services standards such as XML (Extensible Markup Language) and SOAP (Simple Object Access Protocol). These standards, and the Web Services infrastructure that they enable, create possibilities for the Internet's utility as a new source of economic development, business efficiency and consumer convenience.

As the use of Web Services brings benefits to consumers and businesses, the real-time movement of all kinds of information is garnering the attention of IT executives, CEOs and government regulators. As with any new computing paradigm in today's world, the need for information security is receiving a great deal of attention.

This paper distinguishes between the trust and threat security issues that organizations will face as they deploy, and interact with, Web Services. Only by putting in place Web Services threat protection and trust management mechanisms can an organization realistically guard against the impending flood of unauthorized application access and the ever-increasing malicious attacks to information security





TABLE OF CONTENTS

The Convergence of Web Services and Network Security	3
Threat and Trust: The Pillars of Web Services Security	3
Managing Trust Relationships for Web Services	5
The Web Services Threat Profile	6
Dealing with Threats to Shared Content	7
Firewalls Are Not Enough	8
The Inside Threat	9
Changing of the Guard: Protecting the Content Layer	10
The New Content Layer	10
Going Beyond XML Schema for Threat Protection: Web Services Intrusion Prevention	12
Web Services Intrusion Prevention	13
Web Service Security Vulnerabilities and Remedies	14
Putting it All Together: Comprehensive Web Services Security Systems	15
Providing Trust Management and Threat Protection	15

AUTHOR: Walid Negm
Director of Product Management

Forum Systems Inc.

Release Date: 02/15/2004





THE CONVERGENCE OF WEB SERVICES AND NETWORK SECURITY

Web Services are enabling organizations of all kinds to do business, and communicate, as never before. The use of Web Services technologies makes it possible for disparate computer systems to interoperate in a relatively seamless fashion when compared with previous methods of system integration. A byproduct of Web Services' power to integrate systems and applications is that information, often private and confidential, is exposed to those who might maliciously corrupt, or steal, this information. For example, an individual's social security number may be exposed as part of an XML document. Ensuring the integrity, confidentiality and security of content is of paramount importance if the benefits afforded by Web Services are to be realized. Current network security-centric methodologies and tools are inadequate to deal with the new challenges brought about through the use of Web services. IT departments must revisit how the following security tools and techniques must evolve to address the needs of a Web Services-centric world:

► **ACCESS CONTROL** – Traditional network security access control is concerned primarily with discrete applications and systems. The use of Web Services radically changes this, since multiple, disparate applications are tied together – often across organizational boundaries. As such, the universe of potential users is greatly expanded, and extra measures need to be taken to ensure that only authorized users have access to specific content and applications. Access control in the world of Web Services needs to be persistent across time and location, so that the integrity of 'payload' content is preserved.

► **INTRUSION PREVENTION** – Current network security intrusion prevention products deal have no knowledge of the information they carry, focused rather on the transport protocols to ascertain anomalous behavior and network attacks. Since Web Services are dynamic and actionable, malicious users can insert and alter application data, creating fraudulent transactions. The user is valid and the data is valid; however, there is a security breach. Web services intrusion prevention includes forensics, deterrence and monitoring as essential components of threat-side capabilities.

► **FIREWALLS** – As in traditional network security models, firewalls continue to be a critical tool in providing corporate perimeter security. In the Web Services model, firewalls need to focus on inspecting SOAP messages – the XML-based standard interface language of Web services – at the perimeter to ensure that only messages from authorized users are allowed to enter. Firewalls in the Web Services world should also examine WSDL files, as they aggregate Web Services data. Without sufficient perimeter security measures, a hostile user can look into a WSDL file and search for vulnerabilities of Web Service ports, operations and messages.



THREAT AND TRUST: THE PILLARS OF WEB SERVICES SECURITY

With the increase in sensitive information being shuttled over the Internet via Web Services, business and IT executives must address the following aspects of information security:

TRUSTWORTHINESS OF INFORMATION: Trust Management deals with the question, “Can someone be trusted to perform a particular action on a specific object?” Extended enterprises and government agencies are relying on federated (i.e., autonomous, yet collaborative) relationships that cannot ensure the validity of the recipient who uses peer-to-peer, distributed or store-and-forward networks. As such, persistent security mechanisms must travel with information such as documents, messages, or transactions.

THREAT EXPOSURE AND WEB SERVICES: Threat Protection is primarily focused on protecting an organization’s information content from vulnerability to attacks. The threat from internal sources is becoming as great, if not greater, than the threat from outside the organization. “Insiders” can cause serious financial damage in the absence of strict enforcement of information security for those individuals that comprise a virtual organization. The exposure to an information security threat, due to the use of Web Services technologies, forces IT managers to address new questions regarding the reliability and accountability of outsourced business relationships, contract workers or terminated, or disgruntled, personnel.



FIGURE 1: WEB SERVICES SECURITY: A TWO-PART EQUATION.



MANAGING TRUST RELATIONSHIPS FOR WEB SERVICES

Trust is an essential part of business relationships. By managing trust between partners, organizations can ensure a secure information-sharing environment by keeping the “right people” in and the “wrong people” out. Web Services complicates the trust model by making it possible to push information out of the originator’s immediate control as it traverses multiple organizational borders in an extended enterprise.

The traditional network-centric trust model assures the protection of the communication channel between two parties. With Web Services, the emphasis changes to a more content-centric, persistent trust model that secures specific messages and documents irrespective of time and location. In particular, as Web Services becomes the standard format for exchanging information and transactions, enterprises and government agencies are securing specific Web Services content in-transit, as well as at final destination.

The following security services are the cornerstones of establishing trust management for Web Services:

MESSAGE EXCHANGE PRIVACY

Web Services standards, like XML, utilize plain text and therefore explicitly describe a great deal about the content that’s represented in a document. Message exchange privacy assures messages are not readable to anyone except the two parties involved in the exchange of content. Encryption of select content protects the confidentiality of the document until the point of processing, allowing for multiple intermediaries to participate in the flow of information without compromising security.

PAYLOAD INTEGRITY

Due to the store-and-forward architecture of Web Services, the integrity of content at rest, in storage or in processing is of primary concern. Ascertaining payload integrity needs to be a persistent function (not just transport-centric) that allows applications to determine if a document has been altered without proper authorization. By using digital “fingerprints” that are bound to each message, recipients can verify that content is in its original form prior to consuming it.

IDENTITY AUTHENTICATION

Web Services authentication is the process of making sure that the entity (person or machine) requesting the Web Service is really who they claim to be, using evidence such as password credentials. In a Web Services world, where information is being processed across a distributed network, the identity of an individual must be seamlessly shared across multiple security contexts. Using identify authentication that is associated with specific content, as opposed to application sessions, allows decentralized authorities to associate granular security policies across multiple applications.

CONTENT AUTHORIZATION

It is also necessary to control the access to specific resources and content by checking the individuals authorization against an access control list. While traditional authorization is bound to a specific user, Web Services authorization is bound to specific content. This enables security authorization policy decisions to be applied to specific content, or application functionality.



As Web Services dynamically connects business partners, each with their own federated security domains, the precise definition of trust begins to vary as content takes on new meaning within each organization's security context. As such, the use of Web Services technologies creates new opportunities to exploit weak trust relationships. Organizations must now carefully consider how to provide adequate security for their important content in a distributed environment. This new security philosophy requires that IT managers redefine how they deal with a "trusted user" as part of their security infrastructure.

THE WEB SERVICES THREAT PROFILE

Establishing and enforcing adequate security policies requires that IT professionals understand the current security threat profile that faces them. With the introduction of new standards and technologies, constant revision of the threat profile is essential. A typical Web Services threat profile can be derived using a standard Web Services architecture (see Figure 2, below) that highlights a multitude of attack points, potential targets and known weaknesses.

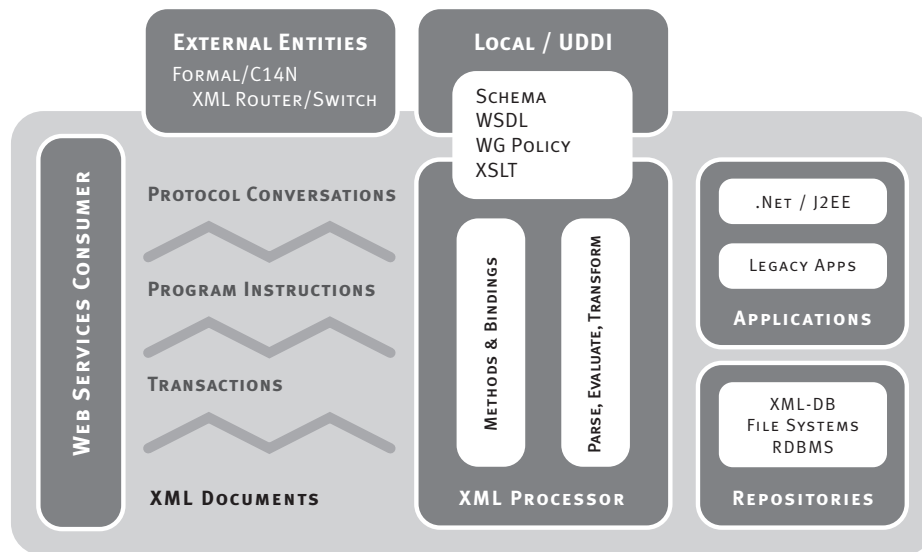


FIGURE 2: WEB SERVICES THREAT PROFILE.

The following describes several areas of potential vulnerability that should be reviewed in the context of developing a threat profile to address Web Services security issues.

XML/SOAP MANIPULATION

XML is the grammar of Web Services, and SOAP is the standard interface language. New implementations, especially when pervasive across applications and entities, are prime targets for potential attackers.

PROTOCOL ABUSE

Web Services utilize higher-level protocols than previous information infrastructure technologies. Given their relatively 'open' nature, each of these protocols provides a set of rules that can more easily be tampered with in pursuit of compromising the security of a system.



COMPROMISED CONFIGURATION DATA / XML STACK / ENTITIES

Configuration data such as XML Schemas and Web Services Description Language (WSDL) files ‘feed’ key information to applications, yet function externally. As such, serious damage can occur if these Web Services technologies are compromised.

XML processors may be standalone utilities, or integrated into any of the components described above. They provide the intelligence to interpret XML documents as inputs to an application. XML processing includes parsing, canonicalization, transformation, data aggregation and XML shredding.

The federation of services, and the malleability of XML documents, creates an environment where a number of entities may participate in providing application functionality. Like a house of cards, a compromised entity may be attacked and bring about the demise of the entire application infrastructure.

The accuracy and integrity of this important information highlights the importance of addressing security deficiencies that could lead to this critical content being compromised.

DEALING WITH THREATS TO SHARED CONTENT

A security threat to an organization’s information systems can occur through unauthorized access to content, malicious modification of content and/or acceptance of erroneous information. Another common threat scenario is the interruption, or prevention, of normal system operation. This type of threat is commonly referred to as Denial of Service (DoS), and can appear as a temporary disruption of a service in order to deny availability of a system to valid users. Such an attack can rapidly degrade business operations, and result in significant unwanted costs to an organization.

Organizations that create a threat profile, specific to their environment, will be better prepared to address the potential security threats that exist. As part of developing an overall security plan organizations should consider the following factors:

- ▶ *AREAS OF VULNERABILITY* – Potential weakness that if exploited, can result in a realized threat.
- ▶ *RISKS* – Exposure points that can result in significant costs (monetary, and otherwise) if a threat is actualized.
- ▶ *ATTACKS* – Specific attempts to exploit vulnerabilities turn a threat into actual damage to an organization’s information systems.

Threat prevention involves security policies and technologies (see Figure 3) to guard, and prevent specific threat profiles from being exploited. This is accomplished by describing and implementing “what actions/conditions may be allowed”, as well as “what is not allowed”. Threat prevention policies, and technologies, protect against well-known and unknown threats using:

- ▶ *PREVENTATIVE MECHANISMS* that control who can use and access specific resources, as well as limiting access privilege to select content and applications.
- ▶ *DETECTION MECHANISMS* that recognize and track attacks using logging, auditing and intrusion detection.
- ▶ *RECOVERY MECHANISMS* that repair damage from attacks, and more importantly, ensure the continuation of normal system activity during attacks.





FIGURE 3: THREAT PROTECTION COMPONENTS.

FIREWALLS ARE NOT ENOUGH

Through the use of Web Services, applications continue to integrate and automate entire value-chains, and have also increased their reliance on information that is distributed in far-flung locations. The bottom-line: information management is more complicated than ever before. In particular, authority to access and act on information is being delegated to the nodes of the application using trust relationships to establish federated privileges. Add an increase in the amount, and type, of content being processed and stored over time, and you have a pronounced need for content security.

Reliance solely on traditional firewall technologies to protect against security breaches from outside, and inside, is no longer sufficient. Companies must look at collaborative information sharing with an eye towards the increased security risk. They must view Web Services threat protection, with more highly evolved, Web Services capable firewall technologies as a means to mitigate this risk. Only by acknowledging the risks inherent through the use of Web Services can IT and business executives begin to put in place the necessary security policies to manage this risk, while at the same time taking advantage of the power offered by Web Services technologies.



THE INSIDE THREAT

The insider attack poses a great challenge to establishing a secure computing environment since the insider with malicious intent has the requisite knowledge to penetrate internal systems. Once authenticated for system access, the internal attacker can expect a high degree of success in carrying out the attack of their choice. For many organizations, the malicious insider may not just be a disgruntled full-time employee. They can be a contractor, business partner, sub-contractor, consultant or even a customer. The threat posed by this type of user, with a motivation to cause harm (financial, political or emotional), is uncomfortable to acknowledge. However, it represents a real danger to the security of an organization's content, and needs to be addressed if security is to be ensured.

On the other hand, an attack from within may not be premeditated, or even intended. It may be the case that an attack by an insider is simply an accident resulting from an experiment gone awry, or from a mistake that is not malicious. The takeaway being, a 'trusted' insider can easily betray their trust without detection, or even intent, if an organization is not alert to this form of threat.

To protect against the abuse, or misuse, of information by an insider, an organization must monitor application usage, detect suspicious transactions, screen information flows and prevent potential corrupt information from moving across the network. By instituting security policies that do not assume guaranteed trust with every electronic interaction, an organization can better protect itself against a new breed of internal information-borne threats that travel across trusted relationships.

CHANGING OF THE GUARD: PROTECTING THE CONTENT LAYER

Today, the main guardians against information security threats are network firewalls. Firewalls monitor and recognize malicious attacks and intrusions that target the network's primary delivery mechanism. They inspect TCP/IP packets and determine the level of threat posed by overall network traffic profiles. Network security products also include virus detection as well as intrusion prevention capabilities that, again, assume the opponent is at the perimeter and working to attack the organization through the network pipe.

Again, Web Services' main asset, and vulnerability, lies in the exposed nature of the information that is carried across business boundaries as multiple, disparate applications are integrated. The 'payload' of the Web Services message should be the focus of security efforts, since this is where the intrinsic value of Web Services is found.

A security breach can occur at any point in the Web Service exchange. For example, the information within a message payload could undergo significant modifications before it arrives at its destination. Simply put, a network view of security must move toward message, or a content-based, view of security (see Figure 4). Tomorrow's 'hackers' will use XML- and SOAP-based messages to deliver attacks that bypass network firewalls. They will be targeting the vulnerabilities associated with Web Services to move harmful content, as well as actionable instructions, directly into and organization's application(s).

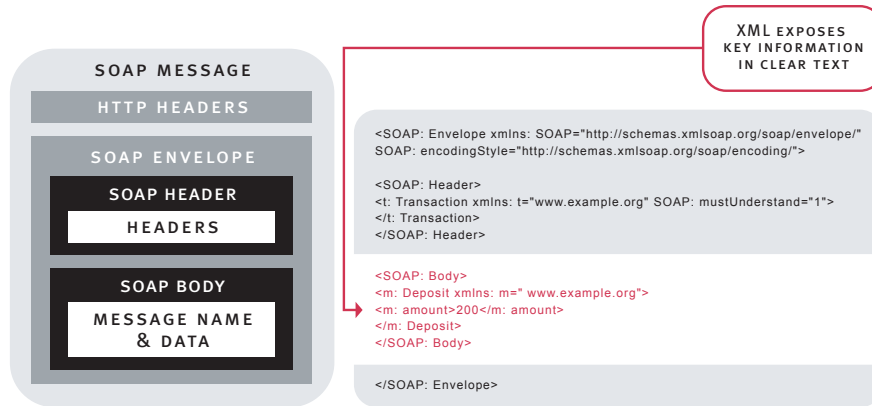


FIGURE 4: XML AND SOAP. Content will harbor new threats.

CONTENT WILL HARBOR NEW THREATS.

THE NEW CONTENT LAYER

SOAP and XML-based messages are the tip of the Web Services standards stack (see Figure 5). There are other important components. The following list identifies core specifications (with associated URL's) that comprise the Web Services platform:

- ▶ XML is the lingua franca of business data representation. XML makes it possible to create self-describing business documents that contain active and context-sensitive information that are processed by machines, applications and individuals. <http://www.w3.org/XML/>
- ▶ SOAP is a protocol specification that defines a uniform way of passing XML-encoded data between two parties, with any number of intermediaries. SOAP provides the 'glue' that enables applications to bind together, and allows the construction of remote procedures and application services, otherwise known as Web Services. The SOAP Envelope, Header(s) and Body constructs deliver content that can change throughout a SOAP document's life cycle. <http://www.w3.org/TR/soap12-part1/>
- ▶ XML Schema provides the mechanism for describing XML message formats, and the constraints on the associated content. XML Schemas allow companies to standardize their data models on industry-specific schemas that define the structure and business rules of the content they exchange with business partners. <http://www.w3.org/XML/Schema>
- ▶ Web Services Description Language (WSDL) is an XML-based document format that provides the "recipe" for Web Services usage. WSDL files are building blocks that facilitates dynamic Web Services interactions between organizations. WSDL is also the mechanism that describes the purpose of a Web Service, as well as how to access and consume it. WSDL files contain both interface and implementation of functionality using SOAP and XML Schema to define the message content. <http://www.w3.org/TR/wsdl>
- ▶ UDDI Directory (Universal Description, Discovery and Integration) allows users (machines or individuals) to locate relevant Web Services over a distributed network. http://uddi.org/pubs/uddi_v3.htm



FIGURE 5: THE WEB SERVICES STACK.

GOING BEYOND XML SCHEMA FOR THREAT PROTECTION: WEB SERVICES INTRUSION PREVENTION

The first line of defense to the Web Service security threats listed above has traditionally been passing Web Services, and the associated content, through structural checks using XML Schemas. An application developer does this by creating an XML Schema to define what can, and cannot, be part of any XML/SOAP message. While this type of message conformance checking may seem to be an adequate security measure to guarantee against unwanted content-borne threats, the reality is, it is an ineffective method of providing adequate security. Content-specific Web Services threats can be grouped into the following categories:

- ▶ **Web Service Denial of Service (DoS) Attacks:** Similar to network packet flooding, Web Service DoS attacks target XML/SOAP parsers to overwhelm them with unsolicited messages that consume CPU, and memory, resources and prevent the servicing of legitimate requests.
- ▶ **Web Service Malicious Commands:** Equivalent to buffer overflow attacks that allow a hacker to control the target system, Web Service Malicious Commands execute application instructions that can query, modify or alter information. For example, a Web Service may be fed hazardous SQL statements. Another Web Service can deliver macrocode to XML-enabled applications, such as Microsoft Office.
- ▶ **WSDL Breaches and Exploits:** This category of Web Services-related threats does not exist in the network packet world, although it is analogous to forced URL browsing and parameter tampering. A WSDL breach can manifest itself as an illegal operation invocation, or modified parameter.
- ▶ **Web Service Viruses:** Similar to launching an executable on a desktop, Web Services- delivered viruses launch known, or unknown, executable code by exploiting weaknesses in XML/SOAP parsers and back-end XML-enabled applications. Web Services-vectored viruses can arrive embedded within the SOAP attachments and pass undetected by basic virus scanning and filtering.



- ▶ **Web Service Worms:** Equivalent to placing self-replicating code on a server, Web Services-delivered worms spread to interconnected ports by exploiting knowledge of WSDL binding information. Web Services-vectored worms may also arrive embedded within SOAP payloads, and pass through undetected by traditional virus scanning software.

The main purpose of XML Schemas is to streamline how business partners integrate their systems together using common taxonomies and vocabularies. For example, insurance Company A would use the same representation for an insurance quote as insurance Company B. Mutual fund trade transactions, between financial services institutions, would be standardized using a financial services-specific XML Schema. True security is beyond the scope of XML Schemas.

SECURITY WITH XML SCHEMAS	
BASIC ASSURANCE:	DO NOT PREVENT:
<ul style="list-style-type: none">▶ Data fields are accounted for (e.g., Invoice Line item and Invoice Header)▶ Data types are checked (e.g., string versus integer)▶ Data formats are checked (e.g., dollars versus euros)	<ul style="list-style-type: none">▶ DoS Attacks▶ Intrusions and Malicious Commands▶ Tampering with WSDL-based Messages▶ Web Services Infected with Viruses and Worms

FIGURE 6: XML SCHEMAS PROVIDE LIMITED SECURITY.

WEB SERVICES INTRUSION PREVENTION

The traditional definition of threat protection is the capability of gathering information about user habits and overall system behavior, using captured anomalous events, and acting on an impending threat. Intrusion Prevention Systems (IPS) unify the concepts of intrusion “detection” and “prevention” under a single framework.

Web Services Intrusion Prevention (WSIP) systems provide Web Services monitoring, vulnerability isolation and proactive threat prevention capabilities that ensure critical systems and applications stay on-line in the face of Web Services attacks. Anomaly detection, signature detection and forensics form the foundation of a WSIP system:

ANOMALY DETECTION

Anomaly detection deals with flagging behavior that deviates from the norm. An example of anomaly detection is, tracking and recoding user access during working and non-working hours. If the system detects that the same user has logged into the system at an unusual hour, say midnight, it will trigger an alarm warning indicating a potential unauthorized intrusion. Select detected anomalies can be added to a signature/pattern repository. An example of a Web Services anomaly detection system is one that distinguishes valid from invalid SOAP messages



by intelligently monitoring Web Services conversations for malicious/unusual behavior. Malicious SOAP/XML messages sent during normal hours are unrestricted. If the same messages defy normal behavior patterns, by either passing through the system at odd hours or if the messages arrive at an unusually fast rate, they are blocked by the WSIP system.

SIGNATURE DETECTION

Signature detection can dynamically look within attack profile databases for known threats, thus, functioning as a digital immune system. An attack against an XML/SOAP parser, or delivery of a malicious SQL command, will have a specific “signature” that will be picked up by the IPS. A Web Services signature detection system would have the malicious command in its database, and detect it before it does damage to a system.

FORENSICS

Forensics establishes facts after the security breach has been committed by performing a post-mortem analysis and investigating the source, and method, of an attack. The forensics administrator reviews the ongoing system logs at the time of the attack using forensics tools to look for clues that will help catch the culprit machine or user that launched the attack. Forensic tools provide insight as to what part of the system the attacker was exploiting to gain entry into the system. In the Web Services world, an administrator would use forensics tools to analyze SOAP/XML messages that triggered intrusion alarms.

VULNERABILITIES AND REMEDIES

Web Services threats can manifest themselves in a number of scenarios with a common premise; if there are vulnerabilities in the security architecture it is likely that someone will eventually attempt to exploit the weakness. The following table lists some of the more significant Web Services security threats that organizations currently face:

CATCH DATA ENTRY ERRORS BEFORE THEY REACH APPLICATIONS	Unless early checks against offending parameter values are performed, applications are unprotected and can crash due to repetitive data validation, and other processing dysfunction(s). Data sanity checks lower costs of fixing data corruption later on.
PREVENT WEB SERVICES FROM BEING EXPLOITED BY DELIVERING CONTENT JUDICIOUSLY	Aggregating data around Web Services allows a malicious user to discover weaknesses in application code. A hostile user can look into a WSDL file and search for vulnerabilities in Web Service ports, operations and messages.
ASSURE GRANULAR ACCESS CONTROL RIGHTS	Traditional access control rights are per application. Since Web Services touch multiple applications, a lack of perimeter accessibility restrictions can lead to “information leakage”.



BLOCK ATTACK ENTRY POINTS	Web Services, by design, create unintended access alleyways, side doors and hidden paths. It is only a matter of time before a malicious user finds a way to bring a service down and prevent access to data.
DETECT ACCIDENTAL DATA ALTERATION	Unless all transactions are scanned for unusual content, erroneous information can cause system inefficiency, or dysfunction (e.g., an erroneous “country” or “state” can cause delays in shipping).
MONITOR AND ACT ON SUSPICIOUS ACTIVITIES	Since Web Services are actionable, malicious users can insert and alter application data, creating fraudulent transactions.
GUARD AGAINST MALICIOUS CODE AND COMMANDS	Hackers can bury malicious SOAP/XML messages inside Web Services and have them slip by network firewalls and application filters. Even SSL-enabled Web Services will unwittingly accepting requests that arrive encrypted.

FIGURE 7: THE URGENT NEED FOR WSIP SYSTEMS.

PUTTING IT ALL TOGETHER: A COMPREHENSIVE WEB SERVICES SECURITY SYSTEM

The need for traffic intelligence, and protection, at the network’s edge is essential as organizations continue to rely on ever-evolving computing paradigms such as store-and-forward messaging, peer-to-peer communication and grid computing. These network computing models are part and parcel of today’s real-world e-businesses, and they are generating information traffic profiles that include new application transport protocols and data/content types. They are also creating a need for more advanced security requirements. The edge of the network must now be capable of enforcing security policies that address a broad set of hybrid traffic profiles, including those resulting from the use of Web Services’ technologies.

Traditional network security solutions will be ineffective against content-borne Web Services security threats. The exposed nature of Web Services technologies, such as XML and SOAP, elevates security issues from the TCP/IP stack to the application layer of the OSI model. Without comprehensive Web Services-capable security in the areas of authentication, firewalls and intrusion prevention, the true benefits of Web Services will be overshadowed by expensive security deficiencies.





FIGURE 8: COMPREHENSIVE WEB SERVICES SECURITY.

PROVIDING TRUST MANAGEMENT AND THREAT PROTECTION

With increasing amounts of sensitive information being vended via the Internet, Web Services security is no longer “nice to have” functionality. It is now a primary business and IT consideration that must be an integral component of IT planning. In order to deliver comprehensive Web Services security, both sides of the trust and threat security equation must be addressed. This need is made more acute by:

- ▶ the increasing demands of recent government legislation (i.e., Sarbanes-Oxley, Gramm-Leach-Bliley, e-Sign and HIPAA); and
- ▶ inadequate protection afforded by traditional security mechanisms, such as packet-based fire walls, transport-centric access control and static intrusion prevention tools.

To address these needs, Forum Systems has developed a complete suite of product to address the ever-changing security needs of the Web Services world. For more information on these products please visit Forum Systems’ website at WWW.FORUMSYSTEMS.COM.