Security Within the Extensible Markup Language Infrastructure

Extensible Markup Language (XML) services (Representational State Transfer [REST], Web services, Electronic Business using XML [ebXML], etc.) continue to gain momentum as the most efficient and flexible mechanism for real-time system-to-system integration. Security for XML services is based firmly on emerging standards defining the application of cryptography, identities, and access control. However, the unique characteristics of XML create uncompromising demands for thorough threat defense against malicious content and XML denial-of-service (XDoS) attacks as well as the need for data security, extensive authentication and authorization, validation, and mediation. XML services infrastructure addresses all these concerns.

The nearly instantaneous, open application integration promised by XML services offers organizations the capability to respond rapidly to new business opportunities. Although this capability is very attractive to forward-thinking businesses, it also presents lucrative opportunities for creative hackers. The business benefits of connecting and automating mutual processes are clear, and XML services technology advances make that these benefits easier to achieve than ever before. However, although direct connections to mission-critical functions improve business responsiveness and results, they also expose the enterprise to a new class of risks: XML threats.

Threat Detection

When assessing the potential threats of Web services, it is useful to look at four major threat categories:

- Message transport security
- XDoS attacks
- Content-based attacks
- People and processes

Message Transport Security

Much of the success of the traditional Web has been facilitated by creation of the Secure Sockets Layer (SSL) Protocol by Netscape and its subsequent broad adoption by all other browser and server providers. The maturity and availability of the technology have enabled many initial XML Web services to use bilateral SSL to authenticate connections and protect against common transport attacks and data compromise.

SSL is a valuable element of an enterprise's XML threat defense framework, but it is insufficient for many XML Web services. The value of SSL is realized primarily when addressing point-to-point sessions. Applications using service-oriented technologies such as Web services and grid computing, which use many transport options or extend beyond a simple point-to-point topology, are not addressed by SSL.

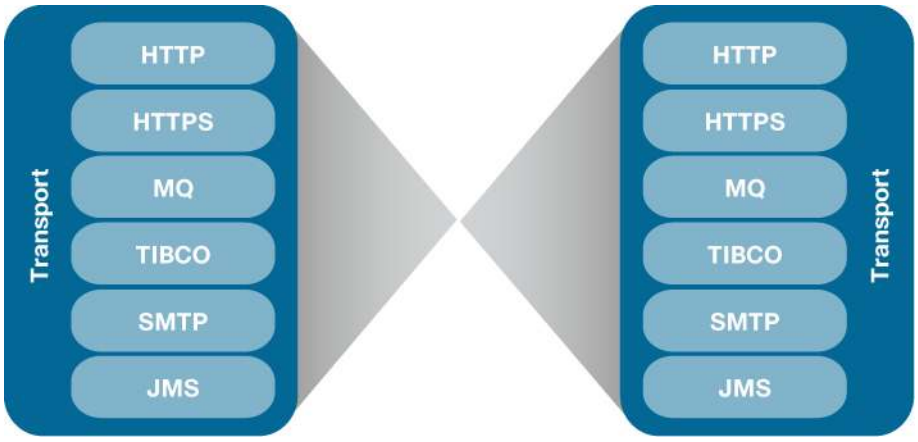
The term Web in Web services misleads many into thinking of client-server approaches based on user interactions using Web browsers. In that model, there are two end points, the browser and the server, and a single transport protocol, HTTP, so a simple point-to-point security model

addressing HTTP works well. In fact, XML Web services operate much more in the manner of a consumer or producer processing model, in which multiple transports may be used and multiple processing steps may be applied to a transaction. This operation creates new opportunities and challenges in creating a practical threat-defense framework.

For example, in addition to HTTP, transports such as the ubiquitous Simple Mail Transfer Protocol (SMTP) (for e-mail) are appropriate for asynchronous or long-lived business transactions and are growing in popularity. SSL does not provide any security support for SMTP or other store-and-forward or message-oriented transports.

XML Web services requirements extend beyond those met by a session-oriented security approach like SSL; the security of the messages or transactions must be addressed directly. Newly ratified standards such as Web Services Security (WS-S) address the broad needs of Web services by directly securing the Web service messages, independent of the transport; message integrity, privacy, and strong authentication for trusted identities are all provided, independent of the security of the underlying transport (Figure 1).

Figure 1. Typical XML Web Services Transport Protocol Mediations



Suggested Tests on the Gateway

- Test the capability of the infrastructure to rapidly provision and debug bilateral SSL.
- Test the performance of the infrastructure in initiating SSL handshakes.
- Test non-HTTP based protocols including SMTP, IBM WebSphere MQ, and Java Message Service (JMS) to determine how the infrastructure enables authentication and secures messages.

XDoS Attacks

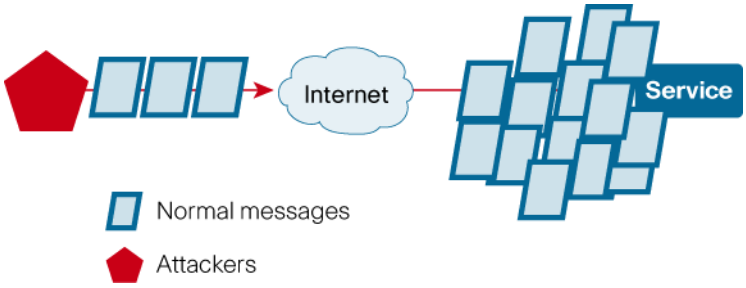
For any network-based service, a DoS attack is a serious and common threat. The fundamental approach used in all DoS attacks is for the attacker to initiate a process on the service provider side that is low cost for the attacker but consumes the resources of the provider to the point where the service becomes inaccessible. SSL itself represents a DoS attack vector, as the overhead of mutual authentication is so high that it could be used by an attacker to consume the service's computing resources (Figure 2).

Although in these early days of XML Web services, it may be easy to discount the probability of XDoS attacks, early adopters have found significant threat from accidental XDoS incidents. Overwhelming a Web service is relatively easy. In comparison to Web servers, which handle

thousands or tens of thousands of transactions per second, Web services tend to handle transactions per second in the tens or hundreds. A partner that simply attempts to interact with the Web service too enthusiastically can accidentally disable the Web service. In addition, the relatively limited experience of developers in creating XML Web services provides opportunities for errors such as infinite loops that can also render the XML Web service inaccessible to all traffic. Existing network or application infrastructure can neither detect nor defend XML Web services from these accidental attacks.

DoS attacks can take place at many points in the network stack. At the data link layer, a huge rate of association requests may cause a wireless access point to fail. At the network layer, a ping attack may bring down a router or server. At the transport layer, fragmentation attacks are possible. In most cases, solutions are available that address the concerns at each of these levels (although the distributed forms of these attacks are still especially troublesome); however, these mechanisms do not address DoS attacks that are launched within the application level, as is the case with XML and Web services. Secure XML infrastructure must provide intelligence through heuristics and alerting to identify patterns that may constitute an XDoS attack against XML and Web services.

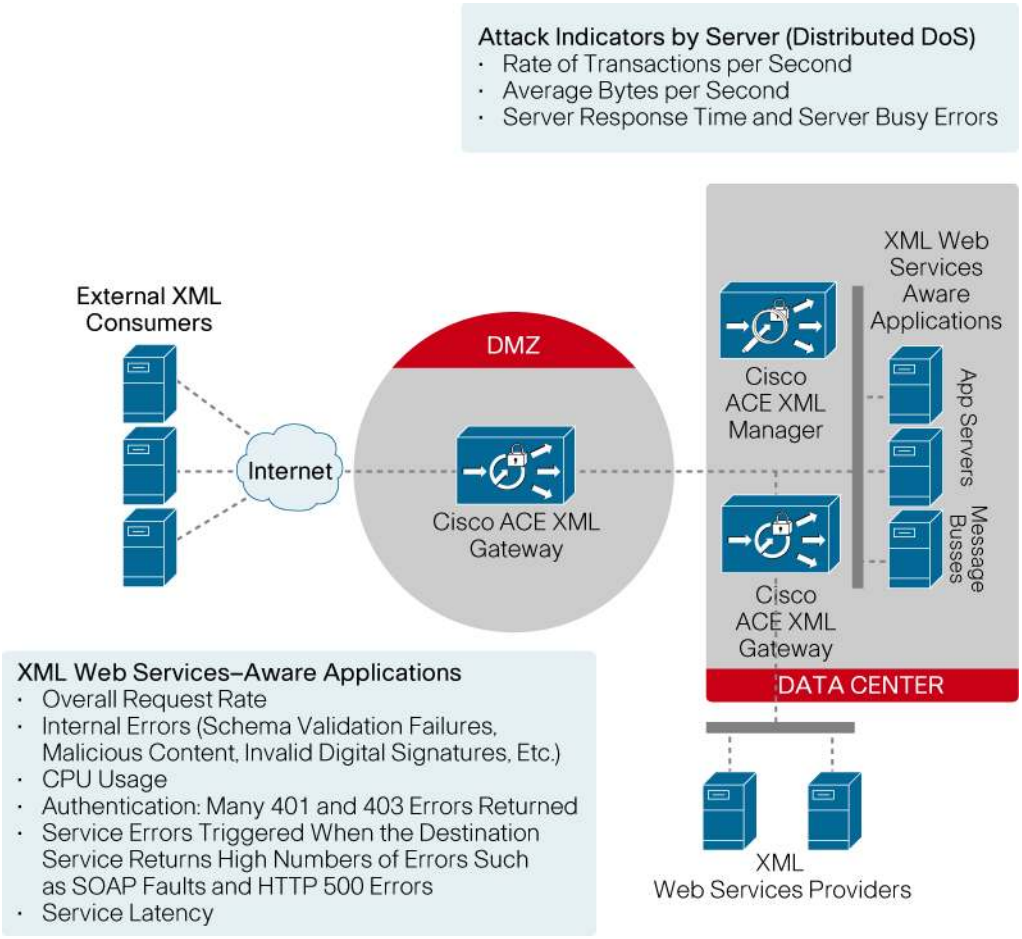
Figure 2. DoS Attack



XML Web services integrate core business applications with each other and business partners. As a consequence, a number of indicators that might be considered an attack are often legitimate business patterns. For example, high message arrival rates may represent an overly simplistic trigger for XDoS detection and avoidance in the case of Web services. Did a business partner trigger a huge transaction load due to the availability of an inexpensive product its Web site was promoting during a major season? This example points to an interesting conclusion about DoS “attacks” as applied to XML Web services: Some may be inadvertent and from trusted sources, but the result is the same. It is therefore crucial to assess XDoS metrics over configurable periods of time and apply appropriate heuristics to classify traffic patterns (Figure 3).

In addition to arrival rates and the flow control of messages passed to back-end servers, the richness and complexity of XML itself provides a unique form of DoS attack vector. Extremely complex schemas or expansion of recursively defined entities represent high overhead processing that ideally suits the needs of an XDoS attacker. In this case, a single XML message employing a technique such as recursive entity expansion can consume all the resources of a service. A huge Simple Object Access Protocol (SOAP) attachment on a small number of messages can have a similar effect.

Figure 3. Metrics to Detect XML Denial of Service Attacks



Suggested Tests on the Gateway

To test XDoS protection on the gateway, perform the following tests:

- Initiate a high number of new connections from the same IP address to a service. Increase the volume of connection requests, messages, and authentication attempts and the message file sizes (1 KB up to 100 MB). How does the gateway convey information about the traffic? Set the gateway to generate SMTP alerts at a threshold and to block IP addresses at a later threshold.
- Initiate a high number of requests to the same service using a wide number of IP addresses to send the requests. Increase the volume of connection requests, messages, and authentication attempts and the message file sizes (1 KB up to 100 MB). How does the gateway convey information about the traffic? Set the gateway to throttle requests based on the number of new connections per second and on the number of bytes per second as well as generating SMTP alerts.
- Change the thresholds for alerts, throttling, and IP blocking and run the tests again to verify that the thresholds can be configured. Also try gradually increasing the volume and bursting the volume to determine how the infrastructure detects and defends against more creative versions of XDoS attacks.
- To test for an XDoS attack embedded in a malicious or ill-formed XML document, such as an entity expansion attack, configure the gateway to limit the size of documents to, for

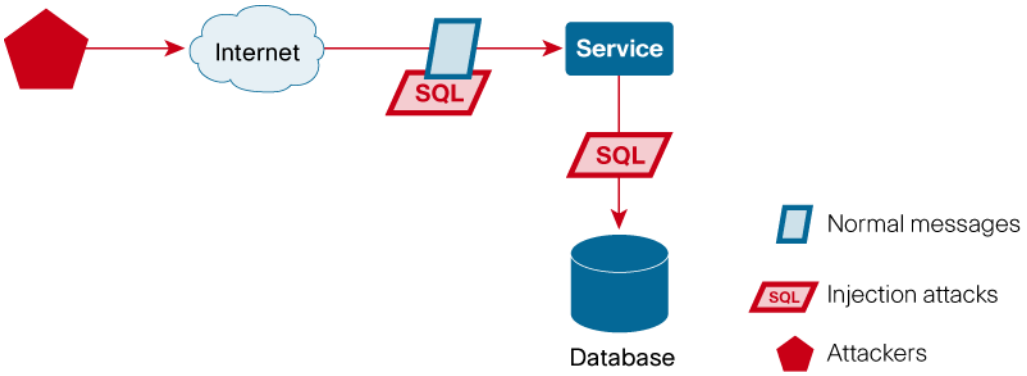
instance, 10 MB; then attempt to send a document larger than that. A more sophisticated test requests an entity expansion, which would result in the document's becoming larger than allowed during processing.

Content-Based Attacks: Protection Against Viruses and Worms

Content-based attacks are threats to the XML Web service that arise from the actual content, or payload, of the XML document (Figure 4). Content is a well-understood attack vector in the HTML browser world, using techniques such as SQL command insertion. Content-based attacks from trusted systems can occur as a result of a user's misconfiguration or compromise of system security in initiating the transaction (for example, infection by a virus).

Content-based attacks are application or platform specific. An application may have vulnerabilities unique to its design. Application-specific vulnerabilities may arise, for example, from the way that the application performs field and type checking when updating database entries. An example of a platform-specific vulnerability is the buffer overflow problem associated with a User Datagram Protocol (UDP) port on a well-known Structured Query Language (SQL) server; the problem was exploited by the SQL Slammer virus.

Figure 4. Content-Based Attack



Because threats are application and platform specific, content threat detection must be specific to the application and platform of the XML Web services. Consequently, a gateway must have the flexibility to configure new filters or to securely upload new filters as new vulnerabilities are discovered in standard platforms or new XML Web services applications are deployed. XML content filtering is more analogous to text filtering than to e-mail virus filtering. For example, in some XML use cases SQL should be accepted, but only specific commands; in some situations, XML may contain the words that constitute a remote procedure call (RPC) statement but which are actually innocuous text.

Types of content-based threats include malicious RPC statements and bad Web Service Description Language (WSDL). Malicious RPC includes SQL injection attacks. Threats can come from the insertion of inappropriate content into a well-formed XML message; such content can be identified through schema validation and detailed content screening filters that are specific to the service and intended connection. In addition, the service WSDL must be checked to help ensure that it is valid and that the external references from the WSDL are valid and available for the production service.

Suggested Tests on the Gateway

To test for detection and defense against content-based attacks, perform the following tests:

- Create a service with a schema that calls for specific data types in specific fields.
- Verify that the security gateway validates schema.
- Send an XML or SOAP message with an invalid function or value specified: for instance, DROP DATABASE.
- Evaluate the gateway's response: both the response to the request and the information captured in the logs for debugging and a secure record of enforcement.
- Send an XML or SOAP message with a valid function call that should be excluded from reaching the service: for instance, a SQL injection or dropping of all tables where the addition of an employee (name and Social Security number) should be.
- Evaluate the gateway's response: both the response to the request and the information captured in the logs for debugging and a secure record of enforcement.
- When specifying a schema to validate or content to filter, take care to avoid specifying too generic a filter and thus preventing legitimate messages from passing through.

The Cisco ACE XML Gateway provides flexibility with two secure and easy ways to add new filters:

- Test the gateway to demonstrate that a new content filter can be added manually through the GUI.
- Test the gateway to demonstrate batch updating by uploading a secure (signed) file.

People and Processes: Threats from the Inside

The gateway should enable an enterprise to implement a policy workflow model consistent with the levels of control required by the business as well as compliance (legal) guidelines. For example, the gateway should support a large number of configurable roles that deliver different privileges to users of the gateway. In addition, the gateway should provide a visual mechanism for comparing proposed policy changes to existing policy so that administrators with approval and deployment rights can easily approve and deny requests and track a workflow of policy development, approval, and deployment. The gateway should also have functions that encourage its use by application developers as they create new XML Web services and provision new connections to existing services; an example is a function that enables an isolated application development team to create services, connections, and policies to test, with the policies easily exported from the team's environment and imported into the product environment for approval.

The XML infrastructure also should provide mechanisms to help ensure that protected services are inaccessible unless the user has been properly authenticated and screened. A signed Security Assertion Markup Language (SAML) assertion inserted on every processed message, with accompanying logic at the service to accept only messages so marked, can be a simple mechanism to thwart insider attacks.

Suggested Tests on the Gateway

To test how the infrastructure encourages and enforces good behavior and compliance, perform the following tests:

- Provision a number of different roles and authenticate using each of them to verify that the gateway properly administers permissions and access.

- Verify that authentication to the gateway can be performed using strong authentication options such as Rivest, Shamir, and Adelman (RSA) SecurID.
- Change an existing policy in multiple places, verify that the approving administrator can see those changes, and test the response when the administrator approves some changes and denies others.
- Test for partial policy deployment support.
- Test for instant policy rollback support (mistakes do happen).
- Log in as a developer and import a new service and create associated policies. Export those policies. Import policies as the security administrator. How are differences and errors identified?
- Attempt to access a service by circumventing the security gateway. Are you rebuffed? How?

Threat Responses

Threat detection capabilities need to be paired with flexible threat-response policies that reflect the security guidelines of the company. In many cases, the response should be an SMTP alert, followed by a logging of the actual message details (not just the event notification itself), so that the operations and development team can assess the threat and respond appropriately. In some cases, automated throttling of the requests to back-end services should be invoked to avoid affecting critical application resources. In extreme cases, automated yet timed IP blocking may be necessary to interrupt an attack in progress without permanently blocking a potentially trusted application or partner.

Conclusion

To help ensure that Web services transactions securely and efficiently reach their intended targets, a dedicated infrastructure that understands XML messages is needed. Gateways provide crucial protection at demarcation points between different trust zones. A gateway must be able to fit transparently with other network infrastructure and be administered by both application and operations staff. It must integrate with existing infrastructure such as directories, public key infrastructure (PKI), and network system management.

Evaluation of the threats to XML Web services must take into account several types of threats, including XDoS and content-based attacks. This evaluation is always application and architecture dependent, and a security gateway must have the flexibility to adapt to evolving threats to critical Web services. In addition, understanding people and processes and their roles in threat prevention will result in a better appreciation of solutions that provide valuable functions such as flexible authentication policies for users and service requestors and roles-based administration for gateway administrators.

Cisco provides the critical XML infrastructure products used by enterprises to realize the promise of Web services. The Cisco ACE XML Gateway enables businesses to secure, implement, and operate XML Web services more efficiently and effectively, accelerating time-to-market for their products and gaining competitive advantages in their businesses. For more information about the Cisco ACE XML Gateway, visit http://www.cisco.com/go/ace.

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C11-410406-00 6/07

