

## **What are the security functions of XML gateways for HTTP (Port 80) and SSL (Secure Sockets Layer) (Port 443) traffic?**

by Gregg Zepp

The growth of web-based applications has led to an application's service or services to function across multiple networks. Many times, these services function across the well-known web traffic ports of 80 and 443. A web service technology such as SOAP travels over HTTP. HTTP is traditionally left open for web traffic at perimeter firewalls. With the advent of Liberty and SAML V2.0's Reverse SOAP (PAOS) specification, SOAP messages can pass through firewalls that limit incoming HTTP traffic but allow outgoing HTTP traffic. Some firewalls have begun to support blocking or allowing SOAP requests based on the source or destination of the request, but more robust and intelligent firewalls are needed to defend networks against malicious SOAP attacks. [1]

Also, because developers often do not practice secure coding standards, this has led to the growth of a number of web application vulnerabilities. Attacks such as Cross Site Scripting and Cross Site Script Forgeries, Injections such as SQL Injection and XML Injection [2], and Insufficient Transport Layer Protection leading to Session Hijacking and Cookie (Session) Tampering attacks have also grown in popularity. [3]

Since many traditional firewalls lack the functionality to sufficiently analyze for and stop malicious web services attacks, a new type of device and service has been evolving: the XML Firewall/Gateway.

The XML Gateway device works as intermediary web service broker. It receives requests from requesters, performs security checks against the requests, and then forwards the requests to an internal Web service provider. From the perspective of the requester, there is only a single provider, in reality there are two. And, there can be any number of intermediary services involved within a single Web service transaction. [4]

Two types of XML Gateway/Firewalls solutions have evolved, non-proxy and proxy based. In proxy based, the connection to the application is controlled by the proxy and no packets or sessions flow to the back-end until the it has inspected and validated the incoming data. Separate TCP sessions are used to manage and inspect user sessions versus back-end server sessions. Non-proxy based either work off a span port by sniffing the traffic or without fully terminating the TCP/IP protocol. These products are often considered an extension of Intrusion Prevention Systems (IPS). [5]

XML gateways are often placed between providers and requesters. An XML gateway often will act as a proxy for the Web service by performing the security-related functionality in its place. Although XML gateways are useful tools in an organization's security strategy, they are not a complete solution. Should an attacker bypass the XML gateway, all internal Web services would be vulnerable to attack. In accordance with a defense in depth approach, all internal Web services must be designed, configured, and coded securely. [6]

An XML Gateway relies on a variety of agreements, protocols, and physical connections. Implementing the XML Gateway involves not only configuring settings on the appliance itself, but configuring settings to accommodate the appliance in the target network (for example, by configuring adjunct firewalls or remote network management devices).

The appliances can be deployed to several types of environments:

- In a production environment, one or more XML Gateways are typically deployed to the network DMZ, often behind a load balancer. In this setting, the Gateway receives client requests from outside the network and passes them back through the load balancer to destination servers within the organization. (See Figure 1)
- During policy development or testing, the XML Gateway usually resides within a protected network.
- The XML Gateway could permanently reside in an internal, protected network, where it is not exposed to external traffic. However, it will need to be able to connect to other network devices to monitor traffic.

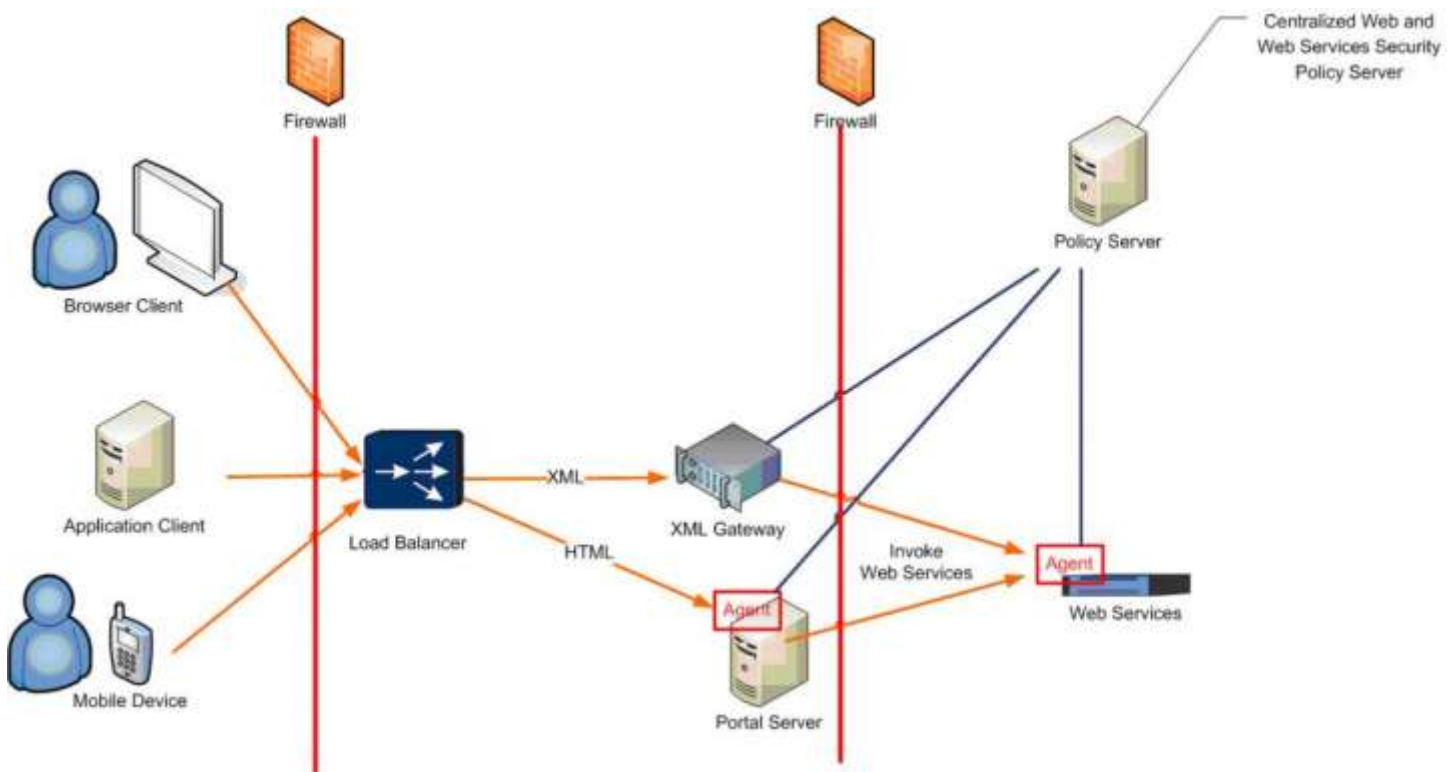
In production deployments, the XML Gateway usually requires access to external networks. Depending on a network topology, one may need to configure an outgoing HTTP proxy for the XML appliance. An XML Gateway uses the proxy for all outbound HTTP connections. The HTTP Proxy settings should be individually configured for the Gateway. [7]

When an XML Gateway or a specialized web service agent is used in combination with the centralized policy server of a web security system, it enables security policies to be configured and managed centrally, but enforced in a highly distributed fashion. In this approach the XML Gateway/web service agent/policy server operates in the same best practice PEP/PDP (Policy Enforcement Point/ Policy Decision Point) architecture that security administrators are so familiar with from their Web Access Management (WAM) products.

In Figure 1, it shows an XML Gateway being used in the DMZ to manage XML traffic coming in from the Internet. Moreover, agents are being used at a Portal Server and at back-end Web services to enforce security policy within those containers. Security policies are being centrally managed with a centralized policy server.

*A Centralized Policy Server* is a central store of security policies that govern the usage of web and web service resources. These policies secure not only HTTP-related traffic but other forms such as SOAP, JMS and MQ using the familiar PEP/PDP security architecture.

In this model, the XML Gateway and the various agents fulfill the role of a PEP using policies from the centralized policy server that acts as the PEP/PDP (Policy Enforcement Point/ Policy Decision Point). However, from the point of view of the XML traffic, this is only part of what the XML Gateway can do. As well as acting as a PEP, the XML Gateway also:



**Figure 1:** XML Gateway deployed to a DMZ

- Performs XML traffic management, including rate throttling
- Transforms XML on-the-fly on the network
- "Enriches" XML by populating it with information sourced from databases, directories, and other XML documents
- Performs protocol mediation, for example, by receiving a message over HTTP then putting it onto an IBM WebSphere MQ queue
- Blocks threatening XML content
- Keeps a log of all Web services usage
- Provides "Swim Lanes" to prioritize traffic for preferred web service clients or customers. [8]

An XML gateway acts as the Web service and forwards all communication to the internal Web service, acting as an intermediary between untrusted services and the internal Web service. XML gateways can provide sophisticated authentication and authorization services, potentially improving the security of the Web service by having all SOAP messages pass through a hardened gateway before reaching any of the custom-developed code. XML gateways can restrict access based on source, destination, or WS-Security

authentication tokens.

XML gateways also support schema validation and some offer support for SOAP intrusion prevention against the following attacks that target vulnerabilities native to XML and XML based services:

- **WSDL scanning.** Attempts to retrieve the WSDL of Web services to gain information that may be useful for an attack.
- **Parameter tampering.** Modification of the parameters a Web service expects to receive in an attempt to bypass input validation and gain unauthorized access to some functionality.
- **Replay attacks.** Attempts to resend SOAP requests to repeat sensitive transactions.
- **Recursive/oversized payload attacks.** Attempts to perform a denial of service against the Web service by sending messages designed to overload the XML parser.
- **External reference attacks.** Attempts to bypass protections by including external references that will be downloaded after the XML has been validated but before its processed by the application.
- **Schema poisoning.** Supplying a schema with the XML document such that the XML validator will use the supplied schema, allowing a malicious XML document to be validated without error.
- **Structured Query Language (SQL) injection.** Providing specially crafted parameters that will be combined within the Web service to generate a SQL query defined by the attacker.
- **Buffer overflows.** Providing specially crafted parameters that will overload the input buffers of the application and will crash the Web service—or potentially allow arbitrary code to be executed.

For Web services that were not implemented with support for WS-Security, XML gateways can be used to act as intermediaries and apply WS-Security to SOAP requests and verify SOAP responses, using HTTPS to secure communication directly between the legacy Web service and the XML gateway. [9]

Web services behind an XML gateway may not need to implement security functionality provided by the firewall, allowing developers to focus only on what the firewall does not support. Because SSL/TLS can be used between the firewall and the Web service, all communication between the Web service and the XML gateway can be trusted. If Web services behind the firewall do not implement security mechanisms to support confidentiality, integrity, and authentication, attackers that bypass the XML gateway may be able to subvert internal Web services. Therefore, it is always beneficial to implement defense-in-depth using XML gateways at the perimeter along with WS-Security or HTTPS for all internal Web services.

Also, XML gateways support in-depth logging facilities for audit purposes. In conjunction with individual audit logs at each Web service. This allows administrators to keep track of what anomalies the XML gateway is experiencing to potentially fine-tune the XML gateway or notice when an attack has been successful and compromised an internal Web service. Nevertheless, the effectiveness of an XML gateway is dependent on the

richness of the feature set and the granularity of policy control. Like any Web service, XML gateways are susceptible to threats from external attackers, so it is important to apply updates and define a policy for handling any intrusions related to it. [10]

### SSL architectural considerations

Application attacks use SSL cryptography and common encoding techniques to hide their attacks and bypass traditional protections. An XML Gateway should be able to decrypt and decode HTTPS sessions. Proxy and non-proxy XML gateways differ in the way they handle SSL cryptography and key management.

Non-proxy units claim to have the technology capable of ‘seeing’ into an SSL encrypted packet as it passes by the non-proxy device. But decrypting and analyzing data requires effort, and by the time a non-proxy unit is ready to make a decision, the attack could reach the back-end servers.

Proxy based units, by contrast, are designed to serve as an SSL termination endpoint. Proxies tightly couple TCP, SSL, and HTTP termination giving them complete visibility into application content and allowing them to perform deep inspection on the entire session payload, including headers, URLs, parameters, and form fields. Some proxy units not only offloads SSL processing, they also can perform certificate based client authentication and SSL re-encryption to the back-end when the security policies demand it. [11]

- 
1. NIST: 800-95: Guide to Secure Web Services. p. 3-32
  2. OWASP - XML Injection: [http://www.owasp.org/index.php/Testing\\_for\\_XML\\_Injection\\_%28OWASP-DV-008%29](http://www.owasp.org/index.php/Testing_for_XML_Injection_%28OWASP-DV-008%29)
  3. OWASP Top 10: [http://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)
  4. NIST: 800-95: Guide to Secure Web Services. p. 2-4
  5. Barracuda Networks: Barracuda Web Application Firewall: [http://www.barracudanetworks.com/ns/downloads/White\\_Papers/Barracuda\\_Web\\_App\\_Firewall\\_WP\\_Benefits\\_of\\_Proxy\\_Based.pdf](http://www.barracudanetworks.com/ns/downloads/White_Papers/Barracuda_Web_App_Firewall_WP_Benefits_of_Proxy_Based.pdf). page 1.
  6. NIST: 800-95: Guide to Secure Web Services. p. 2-11
  7. Planning the Installation of an XML Gateway: [http://www.cisco.com/en/US/docs/app\\_ntwk\\_services/data\\_center\\_app\\_services/xml\\_gateway/v61/administrati on/guide/axg\\_admin\\_planninginstall.html](http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/xml_gateway/v61/administrati on/guide/axg_admin_planninginstall.html)
  8. Unifying Security Policy Across the Web, Web Services, and Web 2.0: <http://www.vordel.com/news/articles/29-10-08.html>
  9. NIST: 800-95: Guide to Secure Web Services. p. 3-43
  10. NIST: 800-95: Guide to Secure Web Services. p. 3-32
  11. Barracuda Networks: Barracuda Web Application Firewall: [http://www.barracudanetworks.com/ns/downloads/White\\_Papers/Barracuda\\_Web\\_App\\_Firewall\\_WP\\_Benefits\\_of\\_Proxy\\_Based.pdf](http://www.barracudanetworks.com/ns/downloads/White_Papers/Barracuda_Web_App_Firewall_WP_Benefits_of_Proxy_Based.pdf). page 3.