What's New in 9.4.2: XML Firewall Features

The introduction of BIG-IP® Application Security Manager (ASM) version 9.4.2 marks a major step forward. BIG-IP ASM now offers more features that are easier to use than prior versions, enabling more granular inspection and policy specification, and helping to maintain its position at the vanguard of Web Application Firewalls (WAFs).

In truth, BIG-IP ASM version 9.4.2 is more than just a WAF. This version of BIG-IP ASM moves toward the concept of Application Delivery Security, enabling any back-end application—not just traditional web applications like most WAFs do currently—to benefit from its protection. Much like the other products in the BIG-IP line, BIG-IP ASM is part of an end-to-end strategy that integrates security into a high-performance application delivery structure. Security is not about the *way* communication occurs with the client, it's about the *data* that goes to the client.

The Application Delivery Security offered by BIG-IP ASM puts the focus on the data itself, regardless of the application that's delivering it. Thanks to the power of TMOS, the Real Time Policy Builder, and, as detailed below, the new XML Firewall features, BIG-IP ASM goes beyond being a traditional WAF and creates a more holistic view of application security[i].

More directly, if it's layer 4-7 traffic, it's an application and BIG-IP ASM can do something to protect it.

What makes this minor-notation ("dot") release seem so major?

While many of the features represent technical or usability advances, this release signifies a fundamental shift in the way BIG-IP ASM interacts with the network, users, and applications. BIG-IP ASM expands its capabilities to that of an Application Firewall. Instead of being limited by notions of what constitutes a web application, BIG-IP ASM examines application traffic for security issues regardless of delivery.

All of the rich features that are part of Web 2.0 sites powered by AJAX (Asynchronous JavaScript and XML), Ruby on RAILS, and JSON (JavaScript Object Notation) technologies now are under the protective umbrella of Application Delivery Security. Applications delivered over the web are not just applications that use the Web 2.0 frameworks; Sales Force Automation (SFA), Enterprise Resource Planning (ERP), and Voice over IP (VoIP) are some examples. Due to their visibility, complexity, and the implementation cost of these types of applications, data security breaches can be a great financial risk for organizations than might come from a "standard" web application.

With this release, the focus is no longer on "positive" and "negative" security models. Those models imply that security breaches can be categorized by simple and static patterns which can be matched or blocked. BIG-IP ASM's focus is on the actions and business logic of applications because there is more risk in *abusing* an application's inherent functionality than there is in *breaking* it outright. Consider online shopping sites; breaking a shopping site is not nearly as damaging or potentially profitable as it is to buy the most popular items for pennies on the dollar surreptitiously.

By taking a wider view of the issues of application security, BIG-IP ASM changes the way it secures applications. At its core, that's what makes this minor release so major.

Application Protection in BIG-IP ASM

As part of striving to protect data regardless of delivery, BIG-IP ASM's capabilities now help to protect against common attacks like SQL command injection and evasion, parameter tampering, and replay ("man-in-the-middle") attacks. One of the largest increases in capability comes in the area of XML filtering and validation. Because XML is used as a data exchange mechanism, it becomes increasingly important to inspect, validate, cleanse, and protect XML transactions.

Following are some of the features BIG-IP ASM has for handling the challenges presented by XML security:

Templates for XML Validation

Featuring more than 20 different profiles for applications, scripting languages, and XML parsers, BIG-IP ASM has the capability to protect applications right out of the box. Delivery mechanisms like RSS feeds, Outlook Web Access, and enterprise back-end applications like Oracle and SAP can now all receive the same protection, ensuring data gets where it needs to go, quickly and securely. Among the included templates are:

- RSS 2.0 Feed Server
- SharePoint 2007
- Outlook Web Access
- XML parsers such as Xerces, libxml2, and Oracle Application Server
- Oracle's Application Portal
- Python, Ruby, JavaScript, and PHP

Ensuring that the XML input and output of web-based applications is both well-formed and only has the required elements becomes increasingly important as more desktop applications can receive and process XML directly. With Outlook 2007, Internet Explorer 7, and Firefox among the applications that can aggregate RSS feeds for the end-user, ensuring that the XML data is "clean" becomes a large part of gaining and maintaining the trust of those users. If users do not trust how and what data you deliver, they will go elsewhere to get what they need, and that's rarely ever good.

Enter BIG-IP ASM's application templates with application-specific XML-security measures. By using these application templates to create an application profile, basic protection for these types of servers and application platforms can be achieved without more than a few clicks of the administrator's mouse. Quickly and efficiently, BIG-IP ASM's templates can help prevent the majority of common XML validation attacks. If the easiest vectors into a system are blocked, most attackers will choose another target. The net result is that applications and data remain secure and available with less effort.

DTD Validation

Building on the existing analysis engine in BIG-IP ASM, version 9.4.2 expands its abilities into the realm of XML content validation against local and remote Document Type Definitions (DTD). By importing DTDs or Web Services Description Language (WSDL)[ii] definitions, BIG-IP ASM can perform validity checks on inbound and outbound XML content in HTTP traffic, blocking unknown or unwanted elements and ensuring that the content is valid based on the schema. Additionally, references to externally hosted schemas can be controlled, ensuring that the schemas processed and validated are from known good sources. Centralizing the control and validation of DTD and WSDL documents reduces the risk to end users if a back-end server is compromised and those documents are modified to point to a malicious site.

Being able to validate XML requests both for proper formatting as well as correct usage of schema elements on requests and responses improves BIG-IP ASM's ability to detect abuse and perform corrective actions. Two types of DTD-oriented attacks are the most common: DTD

redefinition and XXE (XML eXternal Entity). Both of these attacks can cause a client's XML parser, such as the one found in a browser, to come under someone else's control. Because malicious XML grammar could be executed with the same element names as existing legitimate elements, this would bypass any validation in the application itself and leave the client machine open to varying levels of system compromise.

In the case of the first type of attack, the DTD or schema contains a link or other reference that calls a second DTD or schema, which is loaded and processed by the client's XML parser. The client's XML parser then evaluates this second DTD as if it were the initially requested DTD. By exploiting the user's trust of the location of the first DTD, such as an internet shopping site, the second DTD can be loaded, potentially causing problems for the client machine.

Because XML parsers are integrated into many common programs, such as web browsers or word processors, this type of "redefinition" attack can be devastating. The redefinition attack can be countered with a validating type of XML parser. Unfortunately, current versions of Firefox and Internet Explorer do not have an XML parser that validates both DTDs and schema documents. As a result around 93 percent of Internet users are potential targets for this sort of attack, simply because they choose to use either a Firefox or Internet Explorer browser.[iii]

With BIG-IP ASM in the mix, however, all of the validation is done ahead of time, including any link references in the DTD/schema. Any maliciousness present in the DTD/schema is intercepted before it gets to the client's XML parser, preventing problems that could damage user trust in a site or service.

The same goes for XXE attacks. Similar to DTD redefinition attacks, XXE attacks differ because the DTD is simply modified or has additional elements appended to it instead of the wholesale substitution that happens with DTD redefinition. Once again, the validation engine of BIG-IP ASM examines the XML grammar presented by the external document ahead of time, following the HTTP link that's embedded in the original XML grammar, pulling down the external grammar, and then validating it. With the references being checked beforehand, the potential for Cross-Site Scripting attacks in XML documents or even more damaging attacks is reduced immensely.

WSDL Security and Operation Filtering

Simultaneous WSDL documents, like many services-related items, can be a benefit and a hazard. The operations contained in WSDL documents enable the flexible request and retrieval of data as a layer of abstraction from programming language-specific interfaces. With that flexibility often comes the risk of exposing operations used for testing or debugging to the external network, generating the risk for downtime, system instability, or more malicious activity.

Like the aforementioned DTD, WSDLs also need to be validated in similar fashion. Ensuring that there are no problems that could arise from invalid (poorly formed) XML grammar is the first step to securing applications. The second part of securing applications is more difficult because it involves the balance between usability and security.

In the case of WSDLs, some operations that are included in the WSDL may not be for public consumption, such as those that invoke QA test harnesses. There may be other operations that should only be used in certain ways to prevent excessive resource consumption. While it may be tempting to deny external invocation of a site's WSDL, customers and partners may depend on the data retrieved from that interface, so some permissions need to be granted. Application servers such as BEA WebLogic dynamically generate a WSDL definition for each hosted application. To maintain the balance, you need oversight.

Enter BIG-IP ASM to provide Application Delivery Security. Using BIG-IP ASM's interface, specifying valid WSDL operations can be done with simplicity and granularity.

By importing the WSDL directly into BIG-IP ASM, you realize the benefits of centralized management. Centralizing WSDL enforcement ensures the valid WSDL will not be overwritten with a malicious version and the enabled operations remain consistent across all of the back-end servers. When imported, the WSDL file's operations are displayed as checkboxes in the BIG-IP ASM interface. Want to deny access to certain operations? Uncheck them and then apply the profile to the application. It's that easy. Enabling remote users to invoke the operations they need while ensuring that dangerous operations are not exposed or could be enumerated by a remote attacker helps to contain the exposure (risk) of the back-end infrastructure while maintaining the availability of the application and data.

The balance of usability and security is established and maintained: problem solved.

Adjustable Element Validation Parameters

From the description, this feature does not sound particularly groundbreaking. Much like the aforementioned features, this one comes into its own when put into the context of solving potentially high-risk problems: XML Bombs and Transform Injections.[iv]

In brief, XML Bombs exploit rules that require entity references be expanded for evaluation. XML Bombs add extraneous entity entries to an XML document, usually by including a DTD definition with these entity entries before the root element of the document. These entities are often numbered sequentially and are defined and redefined much like variables in programming or scripting.

For example, if the first entity were "foo" and it was defined as "Foo!" the next entity, "foo2" would be defined as "&foo; &foo." Subsequent entities would then be defined based on the expanded value of the previous entity; "foo3" expands to "&foo; &foo; &foo; &foo" when the two "foo2" entities are expanded inside the definition of "foo2." Since the values of the entities grow large quickly as they are expanded ("foo128" would be equivalent to 2×2^{127} "foo" entities), the result is an exhaustion of memory and CPU resources, causing slow performance or application crashes. An XML parser can fall victim to this attack even if it is instructed not to expand entities because entity expansion is a valid and required action for XML documents.

A similar issue that faces XML security is Transform Injections. Like XML Bombs, the end goal is a Denial of Service by consuming memory and CPU resources of client machines. Transform Injections exploit issues that occur when converting XML data to another form for presentation. Because it is rule-based, the eXtensible Stylesheet Language Transformations (XSLT) environment includes flow-control functions as part of its data transformation toolkit. While these flow-control functions can be very powerful for performing iterative and repetitive tasks, such as converting XML data to HTML, they can also be used maliciously to create infinite loops that can tie up resources. Redundant transforms can also be included ("injected") in transformation instructions, causing resource consumption. When the XML parser evaluates these redundant transforms by transforming the original data repeatedly without modifying it, the parser must allocate memory to keep track of each transformation instruction, even if it does not result in an action.

BIG-IP ASM can handle these problems directly. Limits on maximum entity expansion and the number of levels of entity recursion are among the 15+ parameters that can be adjusted in the graphical interface, thereby containing or defusing the effects of an XML Bomb. The depth of child element pairs, the maximum amount of data contained in an element pair, and the maximum size of an XML document can also be set to mitigate the effects of Transform Injection attacks.

Three built-in security levels (high, medium, and low) provide baseline values for the validation parameters, helping to create policies that can be applied immediately to protect applications and data. A more granular policy can be created and applied, either manually or with the Real Time Policy Builder, should it be required.

Conclusion

Though each of the features detailed above were presented as discrete features, the power and flexibility of BIG-IP ASM is demonstrated through application profiles which can combine aspects of the different XML Firewall features. Being able to create a customized policy for any XML-based application and apply it at a central administration point means administrators can now implement security policies at the enterprise level, not just at an application-specific level. Thanks to its location in the network, BIG-IP ASM 9.4.2 has the capability to remove risk from application delivery.

By taking a holistic approach to Application Delivery Security, BIG-IP ASM's features allow for easier creation, testing, and deployment of policies. By applying policies that work straight out of the box or using the Real Time Policy Builder to create a more specific policy, BIG-IP ASM can implement Application Delivery Security for critical applications much faster and more accurately than its predecessors.

Security needs to extend beyond Web 2.0. While Web 2.0 is about new ways to integrate and present data, securing that data also requires an integrated approach, not just a device that focuses on a small part of getting that data to the customer. As part of F5's integrated Application Delivery Security strategy, BIG-IP ASM has the features, flexibility, and power that can take application security to that next level.

---

[i] For additional information on TMOS, the Real-Time Policy Builder, and other aspects of BIG-IP ASM's architecture, please visit www.f5.com.
[ii] For additional information on DTDs and WSDLs, please visit w3.org.
[iii] W3Schools.com, "*Browser Statistics*," July 2007.
[iv] A good overview of XML Bombs can be found at Tech Target's *Search Software Quality* site.