The BIG-IP Application Security Manager Part 5: XML Security

John Wagnon, 2013-21-10

This is the fifth article in a 10-part series on the BIG-IP Application Security Manager (ASM). The first four articles in this series are:

1. What is the BIG-IP ASM?
2. Policy Building
3. The Importance of File Types, Parameters, and URLs
4. Attack Signatures

This fifth article in the series will discuss the basic concepts of XML and how the BIG-IP ASM provides security for XML.

XML Concepts

The Extensible Markup Language (XML) provides a common syntax for data transfer between similar systems. XML doesn't specify how to display data (HTML is used for that), but rather it is concerned with describing data that can be manipulated and presented using other languages. XML documents are built on a core set of basic nested structures, and developers can decide how tags are named and organized. XML is used extensively in web applications today, so it's important to have a basic understanding as well as a strong defense for this critical technology.

The XML specification (described in this W3C publication) defines an XML document to be ***well-formed*** when it satisfies a list of syntax rules provided in the specification. If an XML processor encounters a violation of these rules, it is required to stop processing the file and report the error. A ***valid*** XML document is defined as a well-formed document that also conforms to the rules of a schema like the Document Type Definition (DTD) or the newer and more powerful XML Schema Definition (XSD). It's important to have valid XML documents when implementing and using web services.

Web Service

A web service is any service that is available over a network and that uses standardized XML syntaxes. You've heard of the "... as a Service" right? Well, this is the stuff we're talking about, and XML plays a big role. On a somewhat tangential note, it seems like there are too many "as a Service" acronyms flying around right now...I really need to make up a hilarious one just for the heck of it. I'll let you know how that goes...

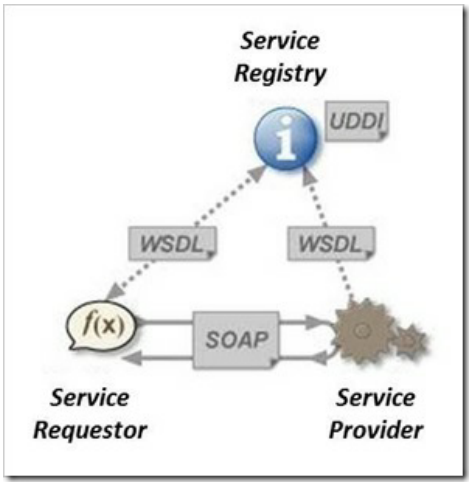
Anyway, back to reality...a web service architecture consists of a service provider, a service requestor, and a service registry.

The service provider implements the service and publishes the service to the service registry using Universal Description, Discovery, and Integration (UDDI) which is an XML-based registry that allows users to register and locate web service applications.

The service registry centralizes the services published by the service provider.

The service requestor finds the service using UDDI and retrieves the Web Services Definition Language (WSDL) file, which consists of an XML-based interface used for describing the functionality offered by the web service. The service requestor is able to consume the service based on all the goodness found in the WSDL using the UDDI. Then, the service requestor can send messages to the service provider using a service transport like the Simple Object Access Protocol (SOAP). SOAP is a protocol specification for exchanging structured information when implementing web services...it relies on XML for its message format. Now you can see why XML is so closely tied to Web Services.

All this craziness is shown in the diagram below. I know what you're thinking...it's difficult to find anything more exciting than this topic!



(Picture copied from Wikipedia)

Because XML is used for data transfer in the web services architecture, it's important to inspect, validate, and protect XML transactions. Fortunately, the BIG-IP ASM can protect several applications including:

- Web services that use HTTP as a transport layer for XML data
- Web services that use encryption and decryption in HTTP requests
- Web services that require verification and signing using digital signatures
- Web applications that use XML for client-server data communications (i.e. Microsoft Outlook Web Access)

ASM Configuration

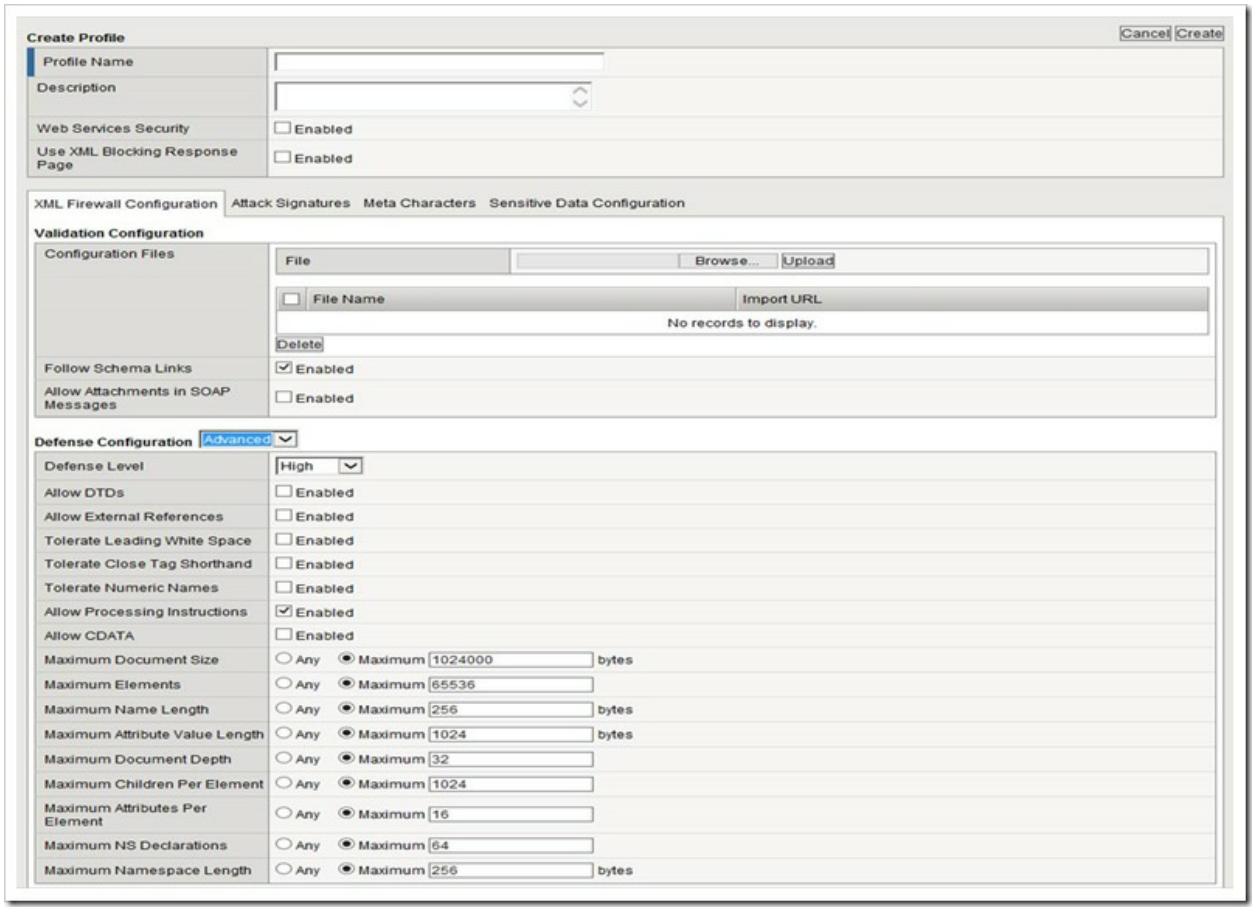
Before you can begin protecting your XML content, you have to create a security policy using the "XML and Web Services" option. After you create the security policy, you create an XML profile and associate it with the XML security policy. You can read more about creating policies in the Policy Building article in this series. To create an XML profile, you navigate to **Application Security >> Content Profiles >> XML Profiles**. When all this is done, the XML profile will protect XML applications in the following ways:

- Validate XML formatting
- Mask sensitive data
- Enforce compliance with XML schema files or WSDL documents
- Provide information leakage protection
- Offer XML encryption and XML signatures
- Offer XML content based routing and XML switching
- Offer XML parser protection against DoS attacks
- Encrypt and decrypt parts of SOAP web services

Validation resources provide the ASM with critical information about the XML data or web services application that the XML profile is protecting. As discussed earlier, many XML applications have a schema file for validation (i.e. DTD or XSD) or WSDL file that describes the language used to communicate with remote users. The XML profile is used to validate whether the incoming traffic complies with the predefined schemas or WSDL files.

The following screenshot shows the configuration of the XML profile in the ASM. Notice all the different features it provides. You can download the all-important configuration files (WSDL), you can associate attack signatures to the profile (protects against things like XML parser attacks -- XML Bombs or External Entity Attacks), you can allow/disallow meta characters, and you can configure sensitive data protection for a specific namespace and a specific element or attribute. Another really cool thing is that most of these features are turned on/off using simple checkboxes. This is really cool and powerful stuff!

I won't bore you with all the details of each setting, but suffice it to say, this thing let's you do tons of great things in order to protect your XML data.



Well, that does it for this ASM article. I hope this sheds some light on how to protect your XML data. And, if you're one of the users who implements *anything* "as a Service" make sure you protect all that data by turning on the BIG-IP ASM. The next time someone throws an XML bomb your way, you'll be glad you did!

Update: Now that the article series is complete, I wanted to share the links to each article. If I add any more in the future, I'll update this list.

1. What is the BIG-IP ASM?
2. Policy Building
3. The Importance of File Types, Parameters, and URLs
4. Attack Signatures
5. XML Security
6. IP Address Intelligence and Whitelisting
7. Geolocation
8. Data Guard
9. Username and Session Awareness Tracking
10. Event Logging

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com