Proceedings of the 10th Colloquium for Information Systems Security Education
University of Maryland, University College
Adelphi, MD June 5-8, 2006

XML Intrusion Prevention
A Comprehensive Threat

Newton Howard and Sergey Kanareykin, *Center for Advanced Defense Studies**

Abstract – This paper describes a comprehensive threat model for a new breed of threats based on XML content, including XML languages used in the Service Oriented Architecture (SOA) paradigm such as SOAP [6] and the Web Services Description Language [11]. In addition to defining a new threat model, this paper compares it to a more traditional network security threat model, by defining it in terms of the network stack. This document also illustrates the concept of XML Intrusion Prevention (XIP) as an analog to traditional network-based intrusion prevention. A new type of threat called an XML Content Attack is also defined, and examples are provided for each layer in the threat model. Finally, this document attempts to use the problem of lost context between XML processing layers to characterize many of the security problems that arise during XML processing.[1]

Index terms – Intrusion prevention, XML, threat model, SOAP, WSDL, XIP, XML Content Attack.

I. XML Intrusion Prevention (XIP)

This work describes a comprehensive threat model for a new breed of threats based on XML content, effectively describing the requirements and scope for XML Intrusion Prevention (XIP). The threat model suggested here considers the sum total of XML processing as a layered architecture that fits between application layer transport mechanisms such as HTTP or SMTP and the application. This new XIP model is required due to the weakest link property of secure systems. In any secure system, an attacker will always try to target the weakest link. As concerns network security, the lower level is a decidedly better understood space, with mature products and technologies available for network intrusion prevention.

However, as the XML processing stack is concerned, this represents a new area with clear weaknesses. These weaknesses arise primarily from the extensibility and verbosity of XML, as well as the requirements for application-to-application communication using the SOA paradigm [1]. Inter-application communication using the SOA paradigm results in XML messages that more closely relate to executable code, opening them up to a wide array of semantic threats. Once XML processing is modeled as a network stack, it begins to inherit many of the security issues of a layered architecture, including the problem of lost context between processing layers. The problem of lost context is a repeated theme within the context of XIP and occurs when a subordinate layer fails to transfer meta-information to an upper layer. In other words, the lower layer does not communicate all of the meta-data required for proper XML processing at the higher layer, resulting in potential vulnerability. Some examples of this condition include the failure to transfer encoding information, object model information, XML security information, or data type information between layers.

II. XML Content Attacks

XML Intrusion Prevention (XIP) is the theory and practice of protecting and mitigating XML Content Attacks. An XML Content Attack tends to be either deliberate or inadvertent. A deliberate attack occurs when any content within an XML document sent to a downstream endpoint puts the endpoint in a state beneficial to an attacker. Conversely, an inadvertent attack occurs when XML content has the same effect

*Center for Advanced Defense Studies (CADS)
10 G Street, NE Suite 610
Washington, DC 20002
Phone: 202-289-3332
Fax: 202-789-2786
newton.howard@c4ads.org
sergey.kanareykin@c4ads.org

as a deliberate attack, but the sender is unaware of the problem and presumed innocent. It may seem unusual to highlight an inadvertent case as a genuine attack, but in principle it is rather simple for a system or entity to generate XML with an oversized payload or incorrect data model.

Not all XML content attacks are weaknesses in the XML syntax itself. In many cases, the weaknesses are a combination of common implementation deficiencies and complex relationships between processing layers. It should be noted that XML Content attacks pervade all languages based around the XML meta-language, including SOAP and XML Web Services technologies [1].

An attacker utilizing an XML content threat may generally rely on at least one of two assumptions:

a. The downstream system parses the XML document using some sort of XML parser. The parser could be custom-built or off-the-shelf.

b. The downstream system is strongly correlating function calls or method invocation to some part of the XML content. That is, the system is effectively executing a portion of the XML document in a loose sense.

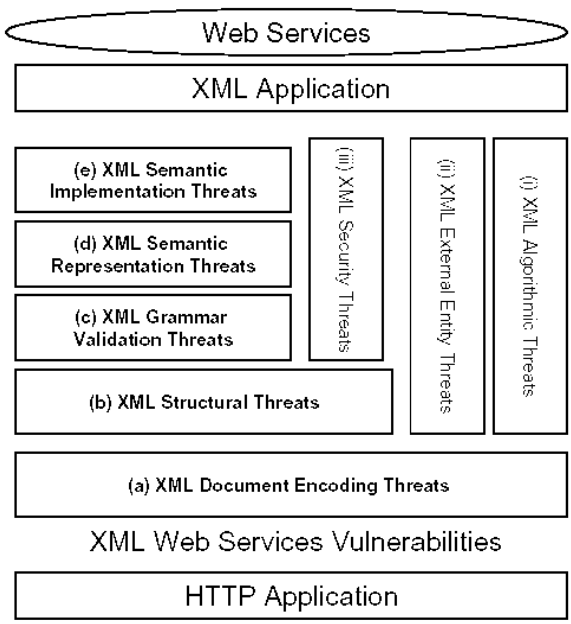
III. The XIP Device

An XIP device is a firewall designed to cover the threats within the XIP space. This section outlines some of the requirements for the XIP device. Any XIP device (XML Firewall) must meet at least two general requirements in order to successfully protect against XML Content Attacks: resiliency and content scrutiny. The resiliency requirement means that the XIP device must be resilient when processing pathological XML data. In other words, the device should not crash, fail, or be driven into an inconsistent state as a result of content contained in the data. The content scrutiny requirement means that the XIP device must not pass along any content that might jeopardize a downstream entity or case any type of unauthorized function call in a downstream entity. In other words, the XIP device must protect itself and all downstream entities that might process the XML content that flows through it.

IV. Layered XIP Model

This section introduces a layered model for describing XML content attacks. The layered XIP model fits into a traditional network model in between application level transport mechanisms such as HTTP and the consuming application. The purpose of the layered XIP model is to create a conceptual structure for the purposes of understanding the XML content attack threat space. The XIP model describes classes of threats specific to XML content and includes threats based around Service Oriented Architectures (SOA) and XML Web Services such as XML over HTTP.

Figure 1: Layered XIP Model



The layered XIP model has two unique properties that differentiate it from a traditional network model: it is naturally recursive, and the threat space is multi-dimensional.

The recursive aspect of the model is driven by the fact that any piece of XML content can be constructed and processed in layers. Each entity that processes a piece of XML content may only process certain nodes or unravel certain parts of the document, leaving further processing to another node. A classical example of this recursive aspect is a SOAP document that travels from one sender to a recipient through intermediaries that may add or remove security properties to the XML document or change the routing headers.

This layered processing implies that the threat model must be fully re-applied to the well-formed subdocument in cases where the XML is not processed in full or by its ultimate receiver. The multi-dimensional aspect of the model refers to

threats that are common to multiple horizontal layers, such as algorithmic threats or xml security threats. The XIP model represents this aspect as a combination of vertical and horizontal threats. In the multi-dimensional model there are specific threats at each level as well as threats that span one or more levels. Horizontal threats are denoted with uppercase letters, while vertical threats are denoted with lower case roman numerals.

In addition to multi-dimensional threats, boundary layer threats that may also occur between horizontal boundaries. These boundary layer threats usually refer to a problem of lost context between layers (such as encoding scheme) which could be the source of an XML content attack.

The practice of defending against XML content attacks begins with understanding the threat space. Each layer of the XIP model refers to a different threat space, categorized by horizontal and vertical threats.

V. Horizontal Threats

Horizontal threats divide the XML payload much like a network stack, where a threat at one level may affect upper layers. Some of the more common horizontal threats include:

a. Document encoding threats

A document encoding threat is any threat that takes advantage of lost encoding information between XML processing layers or imprecise encoding implementations. XML documents support many different encoding types including UTF-8, UTF-16, ISO-8859-1, and windows-1252, and these type differences can introduce subtle holes for an attacker.

b. Structural threats

Structural threats refer to oversized XML components like elements, attributes, comments, or nesting depth. Not all XML parsers behave consistently when handling peculiar well-formed documents, and an attacker may take advantage of an untested code-path within an XML processing engine by exploiting structural threats.

c. Grammar validation threats

Grammar validation is the process of comparing an XML instance to its defining language. The method of grammar validation is variable and can include DTD checking, W3C Schema Validation [2], Relax NG [3], and Schematron-based [4] mechanisms. From an XIP standpoint, three general problems with grammar validation arise. First, grammar validation mechanisms are not complete in and of themselves in fully specifying content models. Second, certain grammar validation mechanisms (such as W3C schema) blur standard object models such as XPath [5]. Finally, many XML language definitions are built to be extensible, providing weak links for an attacker to exploit.

d. Semantic representation threats

Representation threats refer to cases where XML represents remote procedure or document passing calls. In these cases, underlying languages like SOAP [6] or XML/RPC may be subject to semantic attacks where function parameters are altered to induce malicious behavior.

e. Semantic implementation threats

Implementation threats refer to semantic changes within an XML document that rely on the actual implementation consuming the XML document. These cases might use custom implementations, such as a C or Java program, XSLT engine, or Perl script.

VI. Vertical Threats

Vertical threats are pervasive across many horizontal layers and may be seen in more than one point during payload processing, depending on how the XML content model is designed. Some of the more common vertical threats include:

a. Algorithmic threats

Algorithmic threats refer to implementations of standard processing algorithms (such as hash tables) that might be efficient in the average case, but can degenerate with carefully chosen input. For example, a hash table might degenerate from constant time O(1) behavior to linear behavior O(n).If an attacker can guess the data structure or general algorithm used, he or she might be able to produce a Denial of Service (DoS) condition in the XML processing application. While these types of threats are not specific to XML processing, they are important for XIP due to the intense algorithmic processing that XML undergoes due to its highly structured nature.

b. External entity threats

XML documents use URI mechanisms to refer to external or document internal data and structures. These URI references are de-referenced not only during grammar validation, but also during application processing. To this effect, an external entity threat is any type of threat where an external URI is overloaded with malicious data that causes an

XML application to dereference an external source unnecessarily, which might download external code or force a system to waste resources. This threat category covers all types of external entity attacks, not just those based on the Document Type Definition.

c. XML security threats

XML documents may have W3C security mechanisms such as XML Signature [7] or XML Encryption [8] applied, either directly or under the profile of another specification such as OASIS WS-Security [9]. As XML Signature and XML Encryption are in fact XML representations, they have the same inherent security problems as generic XML data, including attacks based on encoding and external entities. Further, specialized attacks such as the Davis Attack [10] are also possible due to the recursive nature of XML Encryption and XML Signature. Any threat that is capable of subverting or manipulation message level security mechanisms without directly attacking the cryptography is considered an XIP-based XML Security threat.

VII. Appendix: Real-World XML Related Threats

The following represent certain documented XML threats encountered by major vendors with deployed infrastructures:

1. *Macromedia JRun and ColdFusion MX*
http://www.macromedia.com/v1/handlers/index.cfm?ID=23559

2. *Sybase EA Server*
http://my.sybase.com/detail?id=1022856

3. *Microsoft .NET Framework 1.1*
http://support.microsoft.com/default.aspx?kbid=826231

4. *Winamp*
http://secunia.com/advisories/12381/

5. *Oracle 9i Application Server & Database Server*
http://www.oracle.com/technology/deploy/security/pdf/2004alert65.pdf

6. *Sun ONE Integration Server EAI Edition*
http://xforce.iss.net/xforce/xfdb/11066

7. *Sun JDK v1.4.2*
http://www.securiteam.com/securitynews/5CP0S0AB5I.html

8. *Internet Explorer 5.5*
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-1325

9. *PeopleSoft PeopleTools 8.1x*
http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21811

10. *Xerces C++*
http://www.securityfocus.com/bid/11312/discussion/

11. *IBM WebSphere Application Server Enterprise*
http://www1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q=PQ81278&uid=swg

12. *Internet Information Server 5.0, 5.1, 6.0*
http://www.cve.mitre.org/cgiin/cvename.cgi?name=CAN-2003-0718 Microsoft IIS Vulnerability

13. *Microsoft Office XP*
http://www.guninski.com/ex$el2.html

14. *Microsoft SQL Server 2000*
http://www.microsoft.com/technet/security/bulletin/MS02-030.mspx

15. *Lib2XML*
http://www.kb.cert.org/vuls/id/493966

VIII. References

[1] W3C Note, Web Services Architecture, 11 February 2004
http://www.w3.org/TR/ws-arch/

[2] XML Schema Part 1: Structures, W3C Recommendation. D. Beech, M. Maloney, N. Mendelsohn, H. Thompson. May 2001.
http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[3] RELAX NG Specification, Committee Specification, 3 December 2001.
http://www.relaxng.org/spec-20011203.html

[4] The Schematron Assertion Language 1.5.
http://www.dsdl.org/0524.pdf

[5] XML Path Language (XPath) Version 1.0. W3C Recommendation. J. Clark, S. DeRose. October 1999. http://www.w3.org/TR/1999/REC-xpath-19991116

[6] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

[7] W3C Recommendation, "XML Signature Syntax and Processing," 12 February 2002.

[8] W3C Working Draft, "XML Encryption Syntax and Processing," 04 March 2002.

[9] A. Nadalin et al., Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard 200401, March 2004.
http://docs.oasisopen.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[10] Donald T. Davis, "Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP,
and XML.", Proc. Usenix Tech. Conf. 2001 (Boston, Mass., June 25-30, 2001)

[11] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, R. Chinnici, M. Gudgin, J-J. Moreau, J. Schlimmer, S. Weerawarana, 10
November 2003.