



NISCC

NATIONAL INFRASTRUCTURE SECURITY CO-ORDINATION CENTRE

**NISCC Viewpoint 06/2006
Issued 21 July 2006**

XML and Web Services

Web services are based on XML. If web services become widely adopted, XML will also be widely adopted. Is this a good thing overall for security? This paper attempts to answer that question, looking at the advantages and disadvantages of XML from a security perspective

NISCC Viewpoint papers are intended to provide an overview of emerging technologies and other issues facing the IT sector. A Viewpoint will not necessarily offer mitigation advice; other NISCC products will do this.

**National Infrastructure
Security Co-ordination Centre**
PO Box 832
London, SW1P 1BG

Tel: 0870 487 0748
Fax: 0870 487 0749
Email: enquiries@nisc.gov.uk
Web: www.nisc.gov.uk

Introduction

1. eXtensible Markup Language (XML) has been the subject of a sustained barrage of hype by many vendors. Despite the hype, it is an important technology which has significantly eased the problems of data exchange and system integration. It is defined by a group of standards that have been promulgated by the World Wide Web Consortium (W3C), and has widespread support from the major IT vendors. Web Services (often also referred to as Service Oriented Architectures (SOA)) are a method of system integration based on the exchange of XML formatted messages between machines. A description of what XML and Web Services are, and some of the major related standards, is included in Appendix A. We assume that the reader is familiar with the basics of XML and the core related technologies in the rest of this paper.
2. The advent of XML and Web Services raises questions about their impact on IT security. This paper will explain some of the advantages and disadvantages of these technologies from a security perspective. First we will describe some generic types of security threat that apply to any system, and then look at how XML can reduce certain of these risks, and finally we will consider the major security problems with XML based systems.

Generic Security Threats

3. All software systems of significant complexity contain bugs. These bugs may be exploited by an attacker to cause the system to behave in ways that are not wanted. In this section we describe two common defects that are particularly relevant to XML technology. A more complete classification of attack vectors can be found on the web¹.
4. Buffer overflows can allow the attacker to corrupt an application by delivering malformed data to it, causing code of the attacker's choosing to run; if the application is running with wide privileges this can lead to complete compromise of the machine. At the very least, a buffer overflow can trivially lead to denial of service. A buffer overflow may occur if data is provided which violates some assumptions (e.g. the size of an array) made by the programmer. Good development processes and testing can reduce the probability of buffer overflows. Rigorous checking of input data before it is presented to the application provides a significant level of defence, so that even if a defect exists in the application itself, attacks will be prevented at the checking stage.
5. Injection attacks attempt to fool the application into performing an action on the attacker's behalf. Common variants include SQL injection, where user originated commands are sent to a relational database, and XPath injection where an application can be tricked

¹ See http://www.webappsec.org/projects/threat/classes_of_attack.shtml

into using data from the wrong part of a document. For example, XPath injection has been used to bypass an authentication system by replacing the real password with a known string.

XML Advantages

6. XML technology has three basic security advantages. It is a standard, it is explicit, and it is checkable.
7. Most of the XML standards have been promulgated by the W3C, although other bodies like OASIS and the UN have been involved in certain areas. Adoption of XML technology has been widespread, because the standards have allowed many software vendors to take it up without having to pay license fees. Most major software vendors use XML in their products, and there are a relatively small number of implementations of the basic software tools required to use XML. These tools are now fairly mature, and bugs or limitations tend to be fixed rapidly, as so much software depends on these basic pieces of software infrastructure.
8. The structure and, to some extent at least, the semantics of data can be recorded in XML. The vocabulary of tags, the rules for how they can be nested, and the definitions of acceptable content can be defined using one of the generic mechanisms like XML Schema. The existence of a generic mechanism for describing data formats means that a large amount of software infrastructure can be reused, rather than re-inventing parsers and verifiers for each new application. Explicit definition of the schema in use allows the sender and receiver of data to agree on what an acceptable data format is, and hence to limit the types of interaction to those that are wanted.
9. Because XML makes the internal structure of data explicit, it exposes the content of the data to much more detailed checking and verification before it enters the critical parts of the system than was typical for systems based on ad-hoc data formats. Detailed checking makes it much harder to inject dangerous content into the systems that use XML data.
10. XML only offers rather limited support for binary data; in general truly binary data like images tends to be encoded into an opaque “blob”², and few of the XML advantages can be bought to bear.

XML Problems

11. Many of the potential problems with XML are simply the failure to exploit the types of verification that become practical, but there are a few new specific problems that its use introduces. Verifying that data conforms to a schema is a relatively computationally expensive operation, and developers often elect to use a non-verifying parser as

² A Binary Large Object. See http://en.wikipedia.org/wiki/Binary_large_object

a performance optimisation. This is very dangerous if the data can come from outside the enterprise as it provides a direct route of attack on the backend systems.

12. The XML standards allow for the use of many different text encodings – these are normally specified at the top of the document. This allows for compatibility with systems that support specific languages by using a specialised encoding; it also allows for the use of one of the Unicode derived encodings which are capable of representing the characters used by most of the world's languages. In several commonly used encodings (including UTF-8, which is now the most widely used encoding) there are multiple ways to represent the same character. Content based checks which do not first transform the text into a standardised representation can be bypassed by a document that hides forbidden content by encoding the characters in a non-standard way; more information is available in our [Secure Web Applications](#) paper.
13. Entities are one of the features that XML inherited from its predecessor technology SGML. They are a form of simple shortcut or macro facility. An entity denotes another block of text which may in turn contain further entities. They can also be used to provide a convenient textual representation for accented or specialised characters. The cascading nature of entity expansion greatly complicates XML content checking.
14. The textual representation used in XML is relatively bulky. This means that it consumes more disk-space and bandwidth than other representations. Disk space is only rarely a problem, but bandwidth is a scarce resource. An XML based interface can be somewhat more vulnerable to denial of service attacks because of the bulk of the representation.
15. The XML standards are sufficiently complex that the software infrastructure that supports them will always contain some bugs. It also means that any checking or verification software is itself too complex to trust unconditionally. In practice, this means that for critical applications, checking by more than one distinct implementation is highly recommended. The complexity also means that the implementations do not always behave in exactly the same way, especially with regard to advanced or less widely used features.
16. HTML³ has a variety of problems that rarely occur with other XML based formats. The most serious of these is the provision for transmitting executable code as scripts or applets. Web browsers attempt to mitigate the risks of running code from potentially hostile sources by using a “sandbox” to run the downloaded code in a controlled environment, but this has historically been a major source of vulnerabilities.

³ HyperText Markup Language (HTML) is the format used for Web pages. It is not strictly XML, but rather conforms to the older and more complex SGML standard. XHTML is a more recent variant which is valid XML.

Web Services

17. "Web Services" refers to a related group of standards that support computer to computer communication for systems integration. These standards are based on the interchange of XML formatted messages. A key difference between Web Services and older technologies for systems integration is that Web Services have been much more widely adopted, and have received support from all the major software vendors.
18. The widespread support for Web Services has encouraged many organisations to use it to solve integration problems that cross the boundary between organisations. Configurations that allow trading partners to exchange data between backend systems are now much more common. This new type of connectivity brings with it a range of new risks. Organisations are now forced to trust not only their own IT security, but also the goodwill and competence of their trading partners. These problems are magnified still further if unknown people can access the interface.
19. All Web Services interfaces are layered on top of an underlying transport protocol. The commonest choice is HTTP⁴, although implementations that use messaging transports like e-mail or IBM MQ-series are also available. Some of these transport mechanisms support additional security controls, to restrict access to authorised users and to ensure the confidentiality and integrity of the data link. For example, the secure variant of HTTP allows the use of digital certificates to identify both ends of the link, and to set encryption for the data. These mechanisms should be used if the interface is accessible over an untrusted medium like the Internet.
20. In the next few years support for a standard called OWL-S⁵ should become available. This is intended to allow the workflow or process model for using a Web Services interface to be described, whereas existing descriptions merely describe the data formats that are passed in and out. This should make it possible to create firewall plugins that can check not only the formats of data, but also only allow the operations that are permitted at that point in the process to be attempted.

Recommendations

21. Applications that use XML based data formats should be designed with input validation. This can either be within the application itself, or

⁴ HyperText Transport Protocol, the underlying mechanism used by web servers and browsers.

⁵ OWL-S is defined by the W3C. It stands for Ontology Web Language for Services. It is currently at Submission stage. Details of the current proposal can be found at

<http://www.w3.org/Submission/OWL-S>

in a separate validator, which might operate as a plugin in the network firewall.

22. Data formats should be tightly constrained. Schema definitions are preferred to the less modern and more limited Document Type Definitions (DTDs⁶). The content types should be clearly defined, with little use of text strings. Text strings that are used to pass through encoded data for other layers of the architecture should be avoided if possible.
23. High risk applications (e.g. those that handle critical data or are connected to untrusted data sources) should ideally have more than one layer of validation using independent implementations.
24. The risks associated with a Web Services interface can be reduced by tightly controlling who has access to it. This should be done at the transport level, as well as any additional controls that are applied at the application level.
25. Web Services interfaces should always be penetration tested before they are opened up to external access. Applications based on these standards are sufficiently complex that even experienced development teams may make mistakes with significant security implications.

⁶ http://en.wikipedia.org/wiki/Document_Type_Definition

Appendix A – Introduction to XML and Web Services

XML

26. XML and related standards such as XML Schema provide a non-proprietary means of defining metadata languages. Metadata applies tags to data so that the structure within a document or message is made explicit. A metadata language describes the metadata tags and their meaning and allows documents or messages to be defined for some specific purpose. Probably the best known metadata language is HTML⁷ whose metadata tags tell web browsers how to display web pages.

27. Below is an example of a fictitious metadata language called QuoteML which describes structured documents containing literary quotations. A fragment of this mythical language might look like this:

```
<quotes>
  <quote author="Descartes">
    <contents>I think, therefore I
am</contents>
    <length>5</length>
  </quote>
  <quote author=...
... and so on
</quotes>
```

28. The XML definition would say that elements are contained between an opening <quotes> and a closing </quotes>. Elements can be nested, for example there are a <contents> and <length> elements within a <quote> element. A tag can have attributed data – in this example <quote> has an attribute author which has a value which is a string “Descartes”.

29. DTDs and Schemas are two ways to describe grammar rules, which can form an enforceable contract between the producer of an XML document or message and the reader. DTDs were invented to support SGML and provide a less rich set of rules than are provided by XML Schema. A document or message that conforms to an appropriate DTD or Schema is called a valid document or message.

30. As an example one could write a DTD for QuoteML that described the following rules:

- The “quotes” element must contain one or more “quote” elements, but nothing else.
- The “quote” element must contain exactly one “contents” element, followed by exactly one “length” element, but nothing else.

⁷ HTML was defined before XML existed and is defined using a predecessor of XML called SGML. There is an XMLised version of HTML called XHTML.

- Inside the “quote” tag itself there must be an attribute “author” which contains text.
 - The “contents” and “length” elements must only contain textual data.
31. XML Schema allow a finer granularity of such rules, but most importantly provide a range of basic types, which can be built up in an analogous way to modern programming languages to form compound types. In the example above one could use an XML Schema to say that the “length” element is a positive integer ... which the definition of QuoteML would then define as the number of words in the quotation contained in the preceding “quote” element.
 32. The basic definition of XML describes a human readable textual language. However to reduce document and message sizes it is possible (but uncommon) to have compact binary encodings for metadata languages defined in XML.

In addition to XML Schema there are many standards⁸ related to XML, such as:

- XSL is a supporting technology that describes how to format or transform the data in an XML document.
- XPath makes it possible to refer to individual components of an XML document. This allows stylesheets in (for example) XSL to dynamically "cherry-pick" pieces of a document in any sequence needed in order to compose the required output.
- XQuery is to XML what SQL is to relational databases.
- XML namespaces enable the same document to contain XML elements and attributes taken from different vocabularies, without any naming collisions occurring.
- XML Signature defines the syntax and processing rules for creating digital signatures on XML content.
- XML Encryption defines the syntax and processing rules for encrypting XML content.
- XPointer is a system for addressing components of XML-based internet media.

Web Services

33. Web Services are a way of implementing the architecture of a distributed computer system. Distributed systems are not new; other methods such as DCOM, CORBA and RMI have been used to implement them, and are still in widespread use. However, Web Services have been gaining ground recently, and are coming into maturity.

⁸ The following list is quoted from the wikipedia (www.wikipedia.org) entry on XML.

34. Web Services are an implementation of service-oriented architecture, where small chunks of functionality with well-defined interfaces are presented as services across a network. The services are presented in a way that other services can invoke them across the network – the “web” in web services is something of a misnomer, as the services are designed to be computer-accessible rather than presented on a web browser. From the point of view of the rest of the network, all that matters about a service is its interface specification. For example – consider a service that provides geographical information. All that other services need to know about it is how to find it, how to structure queries to it, and how to handle replies from it. The actual internal workings of the geographical service are a ‘black box’, and if the implementation of that service changes, then it shouldn’t be relevant to querying the service from the network.
35. One of the most important differences between Web Services and previous incarnations of distributed systems such as DCOM and CORBA is the nature of the standards they are based upon. Previous methods were tightly bound to one infrastructure product or very complex implementations. Web services are at their core based on free, platform neutral, open standards such as HTML, HTTP and especially XML. This isn’t the whole picture as Web Service standardisation has become something of a technical and political battlefield, with different standards bodies having varying ideas on how to go about things. Although some standards (such as XML, SOAP and WSDL⁹) are mature, many areas of Web Service implementation have immature standards. As is often the case, it is not clear whether the best technical solution will emerge from the tensions and standards agendas that currently exist.

⁹ See Wikipedia (www.wikipedia.org) for further details of standards such as SOAP and WSDL.