Mc Graw Hill Education

“本书详细介绍了XML安全性的详细信息和重要内容，可帮助您全面掌握Web服务安全技术。简易的实例和透彻的讨论将使得您的学习更为轻松愉快。”——Colin Adams博士(Webservices.org)

Web Services Security

Web服务安全

技术与原理

- 了解实现Web服务安全的完整过程
- 学习Web服务标准——XML、SOAP、UDDI和WSDL
- 处理XML签名、WS-Security、SAML、XACML和XKMS
- 理解Web服务所面临的挑战和相关法律事项

（美）Mark O'Neill 等著
冉晓旻 郭文伟 译

Mc Graw Hill
清华大学出版社

Web 服务安全技术与原理

(美) Mark O'Neill 等著

冉晓旻 郭文伟 译

清华大学出版社

北 京

Mark O'Neill, et al.
Web Services Security
EISBN: 0-07-222471-1

北京市版权局著作权合同登记号　图字：01-2003-3806

本书封面贴有 McGraw-Hill 公司防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

Web 服务安全技术与原理/(美)奥尼尔(Mark O'Neill)等著；冉晓旻，郭文伟译.—北京：清华大学出版社，2003
书名原文：Web Services Security
ISBN 7-302-07051-2
Ⅰ. W…　Ⅱ. ①奥…②冉…③郭…　Ⅲ. 互联网络—安全技术　Ⅳ. TP393.48
中国版本图书馆 CIP 数据核字(2003)第 070080 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：曹康
文稿编辑：陈宗斌
封面设计：康博
版式设计：康博
印 刷 者：北京密云胶印厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：16.75　**字数**：347 千字
版　　次：2003 年 9 月第 1 版　2003 年 9 月第 1 次印刷
书　　号：ISBN 7-302-07051-2/TP・5183
印　　数：1～4000
定　　价：35.00 元

前　言

本书介绍如何将 Web 服务和信息安全技术相结合。要广泛部署 Web 服务，就必须考虑其安全性问题，而安全技术又需要 Web 服务提供的简易部署的特性，所以 Web 服务和安全性是相辅相成的。至少从理论上看，这是一种双赢的策略。但是该技术非常复杂，并且不易理解。本书将引用大量的实例并通过简洁的语言来帮助您理解 Web 服务的安全技术。

本书首先介绍 Web 服务的概念。Web 服务是一项重要的新技术，以其强大的跨行业支持特性而闻名。IBM、HP、Oracle、Microsoft、Novell 和 Sun 等公司都提供了 Web 服务架构。Gartner 预测：2004 年，Web 服务将在世界财富 2000 强公司的新应用程序解决方案的部署中占据主导地位(Gartner, Inc., 2002)。诸如身份验证、授权和数据完整性等安全概念将贯穿全书。我们首先在前两章中介绍了这些安全概念的含义，以便为后续章节中出现的与 Web 服务相关的内容奠定基础。贯穿本书的线索是 Web 服务安全依赖于这些已确立的安全概念和技术，而这些概念和技术没有什么变化，Web 服务也没有废弃它们。

前两章的内容为讨论与 Web 安全相关的安全协议和过程奠定了基础。接下来的一章不仅讨论了 Web 服务安全的方法，而且还讨论了实现 Web 服务安全的理由。Web 服务安全技术(例如 SAML 和 WS-Security)作用于应用层，但是即便这样，记住 Web 服务完整的安全上下文也很重要。其中包括适当配置的防火墙，使用修补和锁定的 Web 服务器以及(特别是如果使用数字证书的话)适当的安全策略文档。只解决 Web 服务提出的新的安全挑战，而使系统对传统方法的攻击敞开大门，这是一种非常愚蠢的做法。因此，本书除了介绍 Web 服务安全的理论知识之外，还可以用作部署安全的 Web 服务的实用指导手册。这些内容听起来似乎很繁杂，但是一旦您熟悉了基本概念，就很容易理解了。

在接下来的几章中，介绍了一些率先解决 Web 服务安全问题的新技术，包括 XML 签名、XML 加密、SAML、XACML、XKMS 和 WS-Security 等。如果您已经了解了 XML、Web 服务和信息安全的基本概念，那么可以直接阅读这些章节。本书除了讨论这些中立于开发商的技术之外，还讨论了由开发商主导的技术，例如 Microsoft 的.NET myServices 和 Project Liberty。

尽管 UDDI(Universal Description, Discovery, and Integration)的安全领域仍然只是一种假想，并且一直存在争议，但是很多人都对这一争议很感兴趣。因此，本书专门用

一章内容介绍了应用于 UDDI 的 XML Signature 和 XML Encryption 应用程序。我们可以将 ebXML 看作是 Web 服务的替换方案。但是，它包括许多与 Web 服务(XML，当然也有 SOAP)相重叠的技术。此外，ebXML 具有利用 XML 签名和 XML 加密等技术的安全模型。因此，本书还介绍了有关 ebXML 的内容。

与普通介绍技术的书籍不同的是，本书特别利用一章的内容来介绍 Web 服务安全的法律问题。这些法律问题包括应用于 XML 签名的数字签名法律、实现 SAML 时的保密问题、以及由应用程序对应用程序的事务处理所涉及的法律问题。法律问题涉及信息技术的方方面面，Web 服务也不例外。这一章将介绍如何处理如下问题："当某个应用程序连接另一个应用程序，以执行欺诈性的事务处理时，哪一方应该承担法律责任？"以及"不可否认性的切实可行吗？"

尽管本书主要讲述的是如何保护 Web 服务方面的内容，但是您也常常会看到和保护 Web 服务对立的场景——攻击 Web 服务。虽然这些只是假想的攻击，但是了解这些用于攻击 Web 应用程序的技术，并进而将其推广到 Web 服务上并不难。Web 服务安全的未来不仅取决于所开发的针对 Web 服务的攻击，还和所公布的攻击有关。和其他行业一样，"了解您的敌人"非常重要。本书旨在提供保护 Web 服务免遭攻击所需的全部信息。

案例分析附录摘录了本书介绍的 Web 服务主题，并将它们放在实际的环境中。每个案例都是先提出问题，然后再列出解决方案中适合使用的 Web 服务安全技术。记住：必须考虑 Web 服务完整的安全上下文——防火墙、已修正和锁定的 Web 服务器，以及(对于某些解决方案)安全信道(SSL、VPN)或者消息级安全的使用。

本书读者对象

程序员和负责部署 Web 服务的设计人员都必须了解这种新技术对安全的意义。此外，网络安全专业人员必须了解 Web 服务所面临的新的应用层安全挑战，以及解决这些挑战的新的安全标准。公司里实施 Web 服务安全的专业人员和实际实施这些 Web 服务的应用程序专业人员都是本书的读者。

主要读者

本书的主要读者是软件开发人员和实施 XML Web 服务的设计人员。

一般读者

本书的一般读者包括希望了解如何解决由于使用 XML Web 服务而带来的安全漏洞的信息安全专业人员。

本书内容直接明了，尽可能地采用简单的示例，所以普通的非技术型读者也具备了解这个全新领域的相关知识。

目　　录

第Ⅰ部分　导　　论

第 II 部分　XML 安 全

第Ⅴ部分　结　束　语

第Ⅰ部分 导 论

第1章 Web 服 务

当整个计算机行业都赞同一项新技术时，这是一个非常有意义的标志。Web 服务就是这样一种新技术。引领业界的许多公司(例如 IBM、HP、Oracle、Microsoft、Novell 和 Sun)都支持 Web 服务，他们不仅在市场上推广 Web 服务，而且还发布 Web 服务产品。此外，现有的许多产品都将 Web 服务的功能集成到它们的特征集中。所有这些都意味着现在我们拥有大量的 Web 服务平台和可供利用的 Web 服务开发工具。也许会有人指出我们目前真正缺少的正是 Web 服务本身！但是，这些服务正在出现。Web 服务将控制今后几年内新的应用程序解决方案的部署。

Web 服务的许多方面都对安全问题提出了挑战。本书介绍的几种技术可以战胜这些挑战——例如 WS-Security(Web 服务安全性)、SAML 和 XKMS 等技术。本章介绍 Web 服务。Web 服务依赖于 XML，所以本章还将简单介绍 XML 的相关知识。这些内容中包括 XML Schema，由于许多 Web 服务安全规范都包括 XML Schema 的定义，所以理解它非常重要。

已经熟悉 Web 服务的读者可能希望略过本章而直接学习后面章节的内容。但是，如果您对 Web 服务的概念还存在疑问，就应该阅读本章。理解 Web 服务的概念时，初学者所面临的第一个陷阱就是“Web 服务”本身的名称。看到“Web 服务”就猜想(这种方法并非不合理)它必定指的是“Web 上的服务”，这种观念是错误的。事实上，“Web 服务”中的“Web”和“服务”都会使人产生误解。下面就来解释其中的缘由。

1.1 定义 Web 服务

IBM 将 Web 服务定义为“Web 服务是自包含的模块化应用程序，它可以通过网络(通常是指万维网)进行描述、发布、定位和调用。”当定义指的是在万维网上正被调用的 Web 服务时，意味着它们把 HTTP 作为传输层和基于 XML 的消息层的传输协议。但是，Web 服务实际上并不需要 HTTP——可以通过其他传输协议(例如消息排队)传送 XML 格式的数据，该协议更适合一些任务重大的事务。此外，大家都知道万维网和超文本 (当然，特别是 HTML)相关联，而不与 XML 相关联。如果(万一)Web 服务从 HTTP

迁移到其他传输协议，那么“Web 服务”与万维网的相关联程度将减轻，但是它仍然会在其名称中包含“Web”。Web 服务隐含提供“Web”的功能，所以“web”这个词仍然适用。

1.1.1 导航防火墙

Web 服务要想穿过防火墙，通常使用 HTTP 和 SSL 端口(分别指的是 TCP 端口 80 和端口 443)。在“Web 服务”的早期，供应商会说他们的产品是“服从防火墙”的。这意味着防火墙不会阻塞 Web 服务业务，但却会阻塞试图使用 CORBA 特有端口的 CORBA 业务。Web 服务使得在不必打开防火墙端口，或者不必像网络管理员常说的那样“在防火墙中穿一个洞”的情况下部署分布式信息处理技术变得更加容易。这个“躲过防火墙”的部署存在严重的安全隐患。大多数防火墙无法区分 Web 服务业务(通过 HTTP 和 SSL 端口传输)和 Web 浏览器业务。对于一些防火墙来说，有可能同时阻塞 Web 服务业务，但是却不可能为单独的 Web 服务建立不同的规则。本书将在第 3 章中深入探讨这些问题。

1.1.2 面向服务的体系结构：发布、查找和绑定

Web 服务中的“服务”指的是面向服务的体系结构(Service-Oriented Architecture，简称 SOA)。SOA 是分布式信息处理技术中的最新开发项目，在此项目中，应用程序可以通过网络调用其他应用程序的功能。在 SOA 中，功能发布在网络上，因为网络提供了两个重要的能力——“发现”(找到功能的能力)以及“绑定”(连接功能的能力)。在 Web 服务体系结构中，这些活动对应于三个角色：Web 服务提供者、Web 服务请求者和 Web 服务代理者(分别对应于面向服务的体系结构中“发布”、“查找”和“绑定”)。

Web 服务使用动态绑定。这意味着使用 Web 服务的应用程序可以动态设计，并在运行时将客户绑定到服务器上。Web 服务本身使用的编程语言并不重要；作为 Web 服务发布的各种功能的实现可以独立于平台(和编程语言)。如果发布者希望从根本上改变他们实现功能或者发布能被 Web 服务请求者找到(“发现”)的新功能的方式，那么 Web 服务体系结构可以提供这个能力。在 Web 服务出现之前，应用程序通信通常使用静态绑定。静态绑定意味着应用程序的集成缺乏灵活性，因为正在通信的应用程序(以及它们的 IT 部门)必须对使用哪种对象类型和编程语言达成一致。

以前的技术(例如 CORBA、DCOM、Distributed Smalltalk 和 Java RMI)都要求更多的协定，并共享业务系统的上下文。所提供的功能常常被直接链接到用于实现该功能的软件对象中。这并不意味着这些技术是无用的，只是 Web 服务的粗粒度集成可能更

适合于开放的 Internet 上的分布式信息处理。本章引入的许多 Web 服务技术(UDDI、WSDL 和 SOAP)正是为了使 SOA 能够在开放的 Internet 上运行而创建的。

目前我们已经从理论上进行了详细的阐述。下面将了解使用这项新技术的必要因素。Web 服务中的 SOA 发布/查找/绑定功能依赖于 XML。因此，我们首先介绍 XML。

1.2 XML 系列简介

尽管 XML 代表可扩展标记语言(eXtensible Markup Language)，但是缩略词“XML”大多数情况下不仅用于描述 XML 本身，而且还可以描述持续发展的相关技术。核心的 XML 规范本身非常简单。实际上，它是如此简单，以至于许多人第一次看到 XML 时，会很难理解为什么如此简单的技术会有改变世界的如此大的能力。其中的奥秘是不仅要看 XML 语言本身(它只是按照结构化方式描述数据的一种语法)，而且还要看它为了定义和转换 XML、传输并保护 XML 而作用于 XML 的相关技术。这个围绕核心 XML 规范的技术集合才代表了 XML 的真正能力。

1.2.1 XML：定义标记语言的语法

XML 是 W3C(万维网联盟，World Wide Web Consortium)定义的一个规范，它定义了用于定义标记语言的语法。这看起来像是一个不直截了当的定义。XML 定义了使用标记(markup)来组织文档结构的语法。一些专用文档(例如，由 LegalXML 小组定义的法律协议)就是已经用 XML 标记的文档。因为它们使用 XML，所以大量支持 XML 的应用程序都可以使用它们。为了解 XML 的优点，让我们来看一个示例。

1.2.2 结构化文档

有时说每个文档都是一个结构化的文档，这是因为每个文档都有某种内在的逻辑，可以根据这种逻辑将其划分成几个部分，或者将其转换成另一种格式。但问题是要找到这个结构。这样做既费时又费力。过去，通常由应用程序定义它们自己惟一的结构化标记，但是现在的应用程序必须尽可能简便地共享它们的数据，这样，标准化一个结构化标记语言才有意义——这也是创建 XML 的原因。

在过去的 30 年里，人们已经提出了很多结构化数据的方法。其中许多都在商业中使用。例如，EDI(电子文档交换，Electronic Document Interchange)文档主要使用页面的左边距作为结构化文档的方法。下面这行程序(选自 UN/EDIFACT EDI 订货清单)就包

含了供应商的名称和地址：

```
NAD SU   JOE BLOGGS INC      101 SOME STREET      BOSTON, MA 12345      US
```

这个 EDI 文档的结构化规则说明了前三个字符指定这一行中数据的类型。通过分析这个特定类型的 EDI 文档规范，可以发现 NAD 的意思就是名称(Name)和地址(Address)。第 5 和第 6 个字符标识我们正在查看其名称和地址的那个实体的角色——这里指的是供应商(SU 代表供应商 Supplier)。其余部分包含了名称和地址本身，每一行的地址、洲、ZIP 编码和国家代码的位置都是固定的。有关 EDI 文档结构的信息包含在冗长的规范文档中。这些文档说明了各种代码的含义，以及用于提取 EDI 文件信息的偏移位置。

EDI 方法的缺点是缺少有关数据的语义信息。有关 NAD 和 SU 的实际含义的信息包含在一个规范文档中。这也是用逗号分隔的文件的问题。此外，例如那些用在 Windows 初始化文件和 Java 配置文件中的名/值对提供语义信息，但是它们太简单，不能用于复杂的嵌套信息。

XML 根据 SGML 定义了结构化文档的新方法。如程序清单 1-1 所示，XML 中可以提供相同的 EDI 片段。

程序清单　1-1

```
<NameAndAddress Role="Supplier">
<CompanyName>Joe Bloggs Inc</CompanyName >
<AddressLine>101 SOME STREET</AddressLine>
<AddressLine>BOSTON</AddressLine>
<AddressLine>MA</AddressLine>
<ZipCode>12345</ZipCode>
<CountryCode>US</CountryCode>
</NameAndAddress>
```

这个 XML 片段包含的原始数据与相应的 EDI 片段中所包含的原始数据相同，但是它添加了有关数据的描述信息。这些描述信息包含在用尖括号括起的元素(有时称为标记)中。“NameAndAddress”就是这样一个元素，而 AddressLine 是另一个示例。当需要与元素相关的补充信息时，可以将它当作属性添加。角色信息与 NameAndAddress 特别相关，因为它是供应商的名称和地址。地址行嵌套在 NameAndAddress 元素中。

1.2.3　冗长性

从示例中可以看到，XML 非常冗长。W3C(万维网联盟，World Wide Web Consortium)