Web服务安全风险评估研究

摘 要

Web服务是近年提出的一种新的面向Web的分布应用开发与集成框架，它基于面向服务的体系结构，采用Internet通信协议和XML编码传输消息，具有系统平台无关、开发语言无关、松散耦合以及可伸缩等优点，具有极大的发展与应用潜力。尽管Web服务有许多优点，但安全性比较差是其主要缺点，安全性问题已经逐渐引起人们的重视。然而单靠技术不可能从根本上解决Web服务的安全问题，Web服务安全更应从系统工程的角度来看待。而在系统工程中，风险评估占有重要地位，它是Web服务安全的基础和前提，因此，对Web服务进行安全风险评估是十分必要的。

本文分析了web服务的安全技术和web服务安全规范，在详细研究信息安全风险评估相关理论和技术的基础上，对Web服务安全风险评估进行了有益的探索，设计并实现了一个可信的Web服务安全风险评估系统。该评估系统主要从资产评估、脆弱性评估、威胁性评估三个模块展开，用户通过进行资产识别完成资产评估模块；通过完成问卷调查来实现威胁性评估模块；脆弱性评估模块则通过用户在Nessus Client设置扫描规则，由系统调用Nessus Server对目标主机进行扫描，然后将扫描结果进行分析后放入数据库。最后评估系统根据各个模块评估结果算出最后的风险值，并给出建议和改进措施。

最后本文介绍了实验的环境、过程和结果分析。实验结果表明，Web服务安全风险评估系统给出了目标系统的风险等级，实现了预期的设计目标。应用该工具进行风险评估较好地满足了风险评估工具发展的新要求，为用户下一步的Web服务安全风险管理工作提供了科学的有力支持。

关键词： Web服务，Web服务安全，风险评估，Nessus

火龙果•整理
uml.org.cn

WEB SERVICES SECURITY RISK ASSESSMENT

ABSTRACT

Web service raised in recent years is a new integrated frame for the distribution, application and exploitation of the web. Orienting to its service system, web service adopts the Internet communication protocols and XML code transmission message, with the advantages of system platform independent, exploitation langrage independent, loosed coupling and extensibility, etc. It has a great potential for development and application in future.

Although there are many advantages in web service, its low security still came to be the main disadvantage, which has aroused people's attention day and day. However, by technology alone, the security problem can not be solved from the root; much more, it should be treated as a problem of system project, in which risk assessment is of high importance. As the foundation and premise of web service security, it is essential to perform security risk assessment on web service.

In this paper, a useful research on the security risk assessment of web service was done and a reliable web service security assessment system was realized, basing on a careful study of related theory and technology and a precise analysis of the security technology and criterion of web service. This assessment system has three modules, namely, the asset evaluation module, the vulnerability evaluation module and the threaten evaluation module. Asset evaluation is done by user through asset recognition; threaten evaluation is achieved by questionnaire; vulnerability evaluation is performed in such a way that firstly, the user sets scanning formula in the Nessus Client; secondly the system calls the Nessus Server to scan the target main frame and to analysis the scanning result before putting it into the database; finally, the evaluation system will work out the final risk value based on the evaluation value of each module,

and give the user improvement advices.

Finally the environment, process and result analysis of this experiment is introduced in this paper. The result indicates that this security risk evaluation system has provided the risk grades of the target system and has achieved the expected goal. Using this tool in risk evaluation can preferably meet the requirement for the development of risk evaluation tools, and will provide the user with a scientific and strong support to his next work of web service security risk administration.With the maturity of Web services technology and the proliferation of the entire network, Web services security issues have drawn increasing attention. The technology can not fundamentally solve the problem of the security of Web services alone. Web services security should be treated from the perspective of project, and risk assessment occupies an important position in this project, which is the foundation and precondition of Web services security. Therefore, Web services security risk assessment study is very necessary.

KEY WORDS: web services, web services security, risk assessment, Nessus

第一章 引言

1.1 课题背景

1.1.1 Web服务安全现状

目前，Web服务安全采用的是已有的Internet 和WWW的一些安全策略，例如数字签名(digital signature)、XML加密(encryption)技术和标准、访问控制(access control)技术等，然而这些都是静态的措施，并且其本身和Web服务之间就存在技术上的鸿沟。因此，Web服务的安全问题没有得到根本的解决。Web服务是动态变化的，并且不同应用背景的服务所需要的安全措施可能完全不同。现有的一些安全技术只在一定程度上解决了特定系统的安全问题，在Web服务的集成中并不能发挥令人满意的作用 [1]。

目前Web服务中采用的安全技术主要有以下几种：(1) 在客户端建立用户信任机制，执行服务时将相应的认证信息导入服务器；(2) 在SOAP消息头中加入针对特定应用的安全标识(token)，则可从中提取认证、信任消息；(3) 在某个特定的应用领域内，对服务提供者的内部敏感数据进行加密，当其收到服务请求后直接在加了密的数据上进行相应的计算和处理，计算结果解密后返回给服务请求者；(4) 从服务请求者的角度看，请求者需要提交必要的输入数据。考虑到客户信息的安全性，客户将需要提交的服务请求信息进行分块，每次仅提交一个输入数据块，返回的结果对应于该请求，经过多次服务请求，在结果返回之后，由服务请求者进行各次服务执行结果的集成，从一定程度上保证了客户信息的安全。

综上，Web服务安全架构的策略对Web服务系统的设计有较高的要求，并且，Web服务的执行面临着Web服务基本模块与安全模块间的通信、安全模块中的服务信息随Web服务的一致更新等问题，适用于较大规模的Web服务应用，具有高可靠性，但通用性较差；服务加密的方法则侧重对服务本身的管理，可以利用或改进现有的数据加密算法，这种方法可应用到各种Web服务系统中，具有较好的适用性和较低的复杂度，但其可靠性较差[1]。

1.1.2 国内外研究现状

针对Web服务的安全需求，国内外一些标准化组织公司和社会团体都在进行各自的

研究。

W3C（The World Wide Web Consortium）已经制定了一系列的安全规范和标准用于确保信息传递的安全性。其中最重要的有XML加密规范和XML签名规范，前者定义了一种方法用来加密和解密XML文档的全部内容或者部分内容，后者实现了对文档特定部分的摘要和数字签名。另外W3C的另一个工作组还制定了一个密钥管理规范，用于让用户获取加密密钥信息（例如机密密钥证书等）并允许用户进行密钥管理，例如密钥的注册撤销等等。

另一个致力于研究和制定安全规范的团体是结构化信息标准化促进组织OASIS（Organization for the Advancement of Structured Information Standards）。该组织制定SAML语言（Secutity Assertion Markup Language）,为交换身份信息提供了一个框架。另一个著名的标准是XACML（OASIS Access Control MarkupLanguage）,为执行特定操作制定了一个权限集合。

WS-I（Web Service Interaction Interoperability Organization）,是由微软IBM、VeriSign等Web服务投资公司创立，该组织于2002年4月推出了一个Web服务安全白皮书WS-Security。WS-Security描述了为解决Web服务环境中的安全性问题而提议的策略规范，提供了消息完整性和机密性功能。该架构定义了添加到Web服务中所必需的基本元素，并且提供了把安全性令牌关联到SOAP消息的方法。WS-Security为正在提议过程中的Web服务安全性模型打下了基础。随着时间的推移，除安全性以外，这个组织还将引入其它规范来解决安全性策略信任隐私权和授权。IBM Microsoft和VeriSign已经提议把WS-Security作为一种解决Web服务安全性缺陷的方法。

国内对Web服务的安全性研究还处于研究和学习阶段，研究的成果主要是在安全标准的应用方面[3]。

1.1.3 风险评估现状

风险评估（Risk Assessment）是指针对确立的风险管理对象所面临的风险进行识别、分析和评价，即根据风险管理对象的实际环境对它的脆弱性、威胁性进行识别，从而确认其安全风险及其大小[2]。

风险评估最早应用于处于相对独立且严密控制环境下的大型计算机和数据中心。随着个人计算机逐步替代终端系统，以及国际互联网的发展和普及，计算机安全问题日益增加。传统的硬件解决之道，如安装防火墙或自动审计日志对高级管理者来说存在判断困难的问题，并且某些配置并不能有效防止信息泄漏和安全事故的发生。科学、合理的

风险评估为信息安全提供了一种比较根本、有效的解决途径。

风险评估是一项费时、费力的工作，需要专业人员和专业知识的支持。为使风险评估能在各行各业中广泛开展，风险评估工具成为不可或缺的技术支持手段。风险评估工具不仅把技术人员从繁杂的资产统计、风险评估过程中解脱出来，还可以完成一些人力无法完成的工作，如网络或主机中漏洞的搜寻、定位，在历史数据积累和专家知识提炼等方面具有巨大优势，可以极大减轻技术人员的负担，为各种形式的风险评估提供了有力支持。

目前，许多组织根据安全管理指南和标准开发了风险评估工具，为风险评估工作的开展提供了便利条件，但风险评估工具的完善还需要很长一段时间。综观目前各风险评估工具的状况，还存在不少问题，如工具运用的结果如何能够比较准确地反映客观实质、如何有效度量、如何综合和协调工具的使用等。

我国在风险评估工具的开发方面尚处于萌芽阶段，必须大力加强风险评估基础理论的研究力度，积极开发具有自主知识产权的风险评估工具[5]。

1.1.4 风险评估发展趋势

纵观风险评估的发展历程，我们认为，未来在风险评估方面需着重在以下方面展开工作：

（1） 风险评估制度与体系的建立与完善。通过以上对国内外风险评估分析可以看到，在宏观上，风险评估需要一定的依据与规范模式。

（2） 风险评估方法与技术手段的研究。风险评估的实施离不开具体的评估方法与技术手段的支持，正确、合理的方法和技术手段的设计与选择是风险评估准确性与合理性的基础保证。同时，评估过程离不开一定的技术和手段，为了准确、迅速地开展信息安全评估工作，需要开发高效、实用的评估工具。

（3） 风险评估与风险管理的针对性研究。风险评估的目的是为了进行风险的管理和控制，因此，对于评估结果进行管理决策是安全保障的重要内容。根据风险评估结果并采用折中原则建立安全管理体系是风险评估研究工作的最终目的。

1.2 课题的提出

本课题的研究目的就是用风险评估这个新兴的方法学、基础理论和技术来促进互联网上最主要的应用之一Web服务的安全问题的解决，它有着重要理论意义和实践意义。

（1）对Web服务进行安全风险评估是风险评估的应用和扩展。风险评估的理论与方法在其他领域已逐渐发展成一套比较完整的手段和工具，但在信息安全领域，由于其特殊性，仍存在许多困扰管理者和研究人员的问题。我国信息系统的风险评估还处于起步阶段，因此需要从不同的方面和角度来完善风险评估。此课题的研究，可以从Web服务角度来进一步探索和完善风险评估的理论和工作。

（2）对Web服务进行安全风险评估可以科学、高效地解决Web服务不断涌现的安全问题。目前Web服务安全中有很多问题是传统的网络安全技术和安全产品无法解决的，通过对Web服务进行安全风险评估，可以了解Web服务目前与未来的风险所在,从而寻求到一个最佳平衡点，增强Web服务安全可信度。此课题的研究，运用风险评估的方法可以更好地解决Web服务安全问题。

1.3 研究内容

风险评估现已被越来越多的人接受，认为这是一种实现和保证安全的有效且有价值的管理办法。从目前趋势来看，安全风险评估及其分析工具的开发研究将越来越多地从定性向定量转变。本文针对Web服务研究Web服务安全风险评估系统，主要研究工作如下：

（1）研究建立内容丰富的风险评估信息库，包括资产及资产影响库、脆弱点库，威胁性调查问卷库，涵盖多类风险要素属性。

（2）研究Web服务安全风险评估总体方案，提出针对Web服务安全风险评估特点的风险评估流程和风险评估算法。

（3）研究Web服务安全技术和Web服务安全规范，形成科学合理的调查问卷，通过问卷调查法对Web服务风险要素属性进行赋值，实现定性与定量两种方法的统一。

（4）研究漏洞扫描工具Nessus，并对扫描结果进行进一步的分析和模糊处理，客观反映系统的安全状态。

（5）根据研究方案编程，设计实现Web服务安全风险评估系统，并实验测试Web服务安全风险评估系统的可行性。

1.4 论文章节安排

本文的主要内容安排如下：

第一章介绍了Web服务安全现状和国内外研究现状；讲述了风险评估现状和发展趋势；说明了Web服务安全风险评估研究的目的和意义。

第二章介绍了Web服务的定义、Web服务协议栈及其关键技术；对Web服务的安全需求和安全技术进行了研究分析；介绍了Web服务安全规范；详细分析了Web服务主要安全规范和WS-Security规范。

第三章介绍了风险评估的相关概念，风险评估的必要性和风险评估原则；从资产、脆弱点、威胁性三个方面对风险评估流程和算法进行研究分析；最后详细介绍了风险评估的常用方法并对各种方法进行了比较。

第四章在前两章分析介绍Web服务安全和风险评估相关理论的基础上，对 Web服务安全风险评估总体方案进行了设计。提出Web服务安全风险评估系统的设计目标、体系结构和评估流程；介绍了Web服务安全风险评估总体方案的数据库结构模型；最后阐述了风险值的算法。

第五章主要是根据第四章对系统的分析和设计，依据前面的设计结果用软件的形式来实现Web服务安全风险评估系统。详细指出了此系统的开发环境和编程工具，并从风险评估系统各个模块详细叙述了其软件实现，最后通过实验分析实验结果并证明Web服务安全风险评估系统的有效性。

第六章对本文的工作进行了总结，并对未来的研究发展进行了展望。

第二章 Web服务安全机制

本章介绍了Web服务的定义、Web服务协议栈及其关键技术；对Web服务的安全需求和安全技术进行了研究分析；介绍了Web服务安全规范；详细分析了Web服务主要安全规范和WS-Security规范。

2.1 Web服务概述

2.1.1 Web服务的定义

以下是不同组织对Web服务的定义：

定义一：Web服务是自包含的、模块化的应用程序，它可以在网络（通常为Web）中被描述、发布、查找和调用。

定义二：Web服务是基于网络的分布式的模块化组件，它执行特定的任务，遵守具体的技术规范。这些规范使得Web服务能与其它兼容的组件进行互操作。

定义三：所谓Web服务，它是指企业发布的完成其特别商务需求的在线应用服务，其它公司或应用软件能够通过Internet来访问并使用这项应用服务。（UDDI规范2.0）

从以上定义可以看出Web服务有两层含义：其一是指封装成单个实体并发布到网络上的功能实体；其二是指功能集合体被调用后所提供的服务。简单地讲，Web服务是一个URL（Uniform Resource Identifier）资源，客户端可以通过编程方式请求得到它的服务，而不需要知道所请求的服务是怎样实现的，这一点与传统的分布式组件对象模型不同[6]。Web服务的出现，解决了不同平台和异构环境交互的难题，为软件应用系统集成，网格计算等技术的发展注入了活力。

2.1.2 Web服务的体系架构

文献[12] [13]从面向应用的角度，描述了Web服务的基本架构。该架构由3个参与者和3个基本操作构成。3个参与者分别是服务提供者（Service Provider）、服务请求者（Service Requestor）和服务代理（Service Agent），而3个基本操作分别为发布（Publish）、查找（Find）和绑定（Binding）。Web服务基本架构如图2-1所示。服务提供者将其服务发布到服务代理的一个目录上；当服务请求者需要调用该服务时，他首先利用服务代理提供的目录去搜索该服务，得到如何调用该服务的信息；然后根据这些信息去调用服务提供

者发布的服务。当服务请求者从服务代理得到调用所需服务的信息之后，通信是在服务请求者和提供者之间直接进行，而无须经过服务代理。Web服务体系使用一系列标准和协议实现相关的功能，例如：使用WSDL（Web Service Description Language，Web服务描述语言）来描述服务，使用UDDI（Universal Description Discovery and Integration，通用描述发现和集成）来发布、查找服务，而SOAP（Simple Object Access Protocol，简单对象访问协议）被用来执行服务调用[14]。

在Web服务架构的各模块间以及模块内部，消息以XML格式传递。其原因在于，以XML格式表示的消息易于阅读和理解，并且XML文档具有跨平台性和松散耦合的结构特点；从商务应用的角度看，从工作流到查询数据库，直到同贸易伙伴交换信息，XML格式表示的消息封装了词汇表，可以同时在行业组织内部和外部使用；它还有较好的弹性和可扩展性，允许使用附加的信息，并且XML标签提供了可访问的进程入口，从而可强化商业规则，并且增强了互操作性，为信息的自动处理提供了可能[1]。

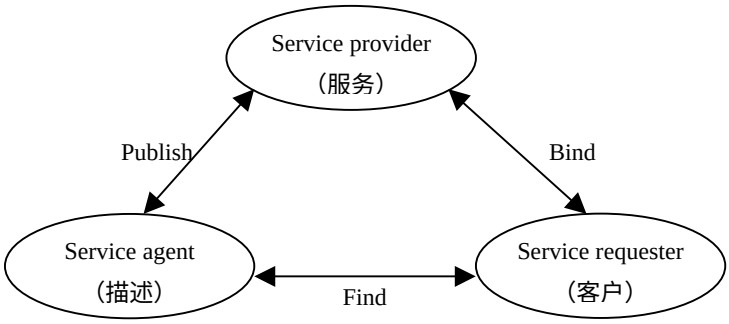

图 2-1 Web 服务体系架构

Figure 2-1 Architecture of Web services

2.1.3 Web服务协议栈

Web服务技术核心的基础是一组基于XML的开放标准协议，这组协议之间具有一定的层次关系，可以用协议栈描述。图2-2描述了Web服务协议栈的层次结构：每一个横条表示协议栈的一个层次，上面的层次是建立在下面各层所提供功能的基础之上，垂直条表示协议栈的各层都需实施的要求。左边的文字表示协议栈各层现有的技术标准。

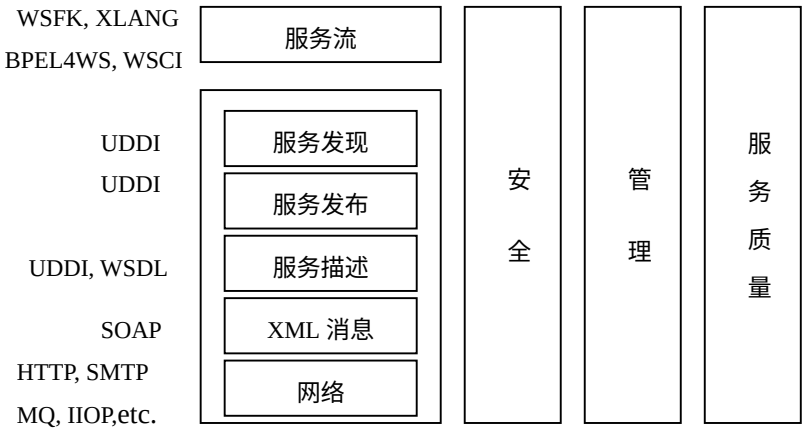

图 2-2 Web 服务协议栈

Figure 2-2 Web services protocol stack

网络层提供传输功能，XML消息层提供远程调用标准，服务描述层提供接口描述标准，这三层是Web服务平台的基础。目前绝大多数Web服务，把HTTP作为基本的网络传输协议，SOAP作为XML消息机制和WSDL作为服务描述标准。服务发现和服务发布是紧密联系的，UDDI提供了一个统一的服务描述、发布、发现和集成机制。组合Web服务可以生成新的更高层的Web服务，服务流可以组合商业过程中各个相当独立的Web服务实现商业流程。为了适用于现实的B2B电子商务环境，Web服务必须支持安全、管理和服务质量等要求[8] [34]。

2.1.4 Web服务关键技术

XML是所有Web服务标准的基础，它是W3C认可的可用于结构化数据和内容以及交换电子文档的标准数据格式。XML广泛用作标记语言的事实标准，通过Internet在应用、系统和设备之间交换信息。在Web服务架构模型中，XML起着至关重要的作用，它是各种通信的通用格式，表示复杂的数据结构。

SOAP是在分布式环境中通过XML交换信息的一种简单协议。SOAP完全继承了XML的开放性和描述可扩展性。SOAP使用基于TCP/IP的应用层协议如HTTP/SMTP/FTP等可以与现有的通信技术最大程度地兼容。SOAP为使用XML在松散分布的环境中对等地交换结构化信息提供了简单的机制。SOAP本身并不定义任何应用语义如编程模型或特定语义。一个完整的SOAP消息包括一个SOAP信封和零个或多个附件，其中SOAP信封包括报头项和SOAP主体，报头可以有一个或多个，可以在报头项中实现自定义的操作。Web服务安全操作是通过对报头项添加安全项实现的。SOAP主体是信息的载体，传输请求的数据。

WSDL是描述Web服务的XML格式语言。WSDL把Web服务定义为网络端点的集合，它有一个根元素用类型、消息、端口类型、绑定、端口和服务等元素来定义Web服务。其中类型是消息的数据类型定义，通常用来描述交换消息；消息代表待传输数据的抽象定义，由一个或多个部分组成；端口类型表示抽象操作的集合，绑定使操作和消息的具体协议与数据格式规范关联；端口指定一个用于绑定的地址，由此定义一个通信端点；服务则是相关端口的集合，在WSDL中端点和消息的抽象定义与具体的网络布置和数据格式绑定，是相互分离的，这样就可以抽象定义消息和端口类型实现它们的重用[15]。

UDDI是一套面向Web服务的消息注册中心的实现标准和规范。创建UDDI注册中心的目的是实现Web服务的发布金额发现利用UDDI规范[16]。在Web上建立发现服务，这些发现服务为所有请求者提供了一致的接口使得已经发布的Web服务能通过编程被需要的请求者发现。UDDI规范文本定义了UDDI操作入口能够支持的API接口。其中用XML描述的数据结构具体定义UDDI。注册中心是对所有提供公共UDDI注册服务站点的统称，在逻辑上它是一个统一体，但在物理上则以分布式系统架构实现，不同的站点之间采用对等网络结构实现，因此访问其中任意一个站点就等同于访问UDDI注册中心，一般情况下访问UDDI操作入口站点所获得的结果是整个UDDI注册中心所覆盖区域的信息。信息查询无需身份认证但必须通过UDDI操作入口站点，自身的用户才能在该站点上进行信息发布，同时以后的更新、删除都必须通过这个操作并使用初始发布时使用的用户名进行权限认证[9]。

2.2 Web服务安全分析

2.2.1 Web服务安全需求

安全包括以下七条安全性原则，这些原则作为一个良好安全性解决方案的基础已经被普遍接受。

(1) 认证。认证是指对实体所宣传的身份确定其有效性。那些请求访问某个受保护组件的请求发起者，或者是某个安全会话或事务处理的发起者，必须提供能够证实其身份的凭证信息。

(2) 授权。授权也称为访问控制，它是指根据属性、断言或相关环境来确定某个已被证实身份的实体是否被允许访问某个受保护的资源。属性是名称-值（name-value）对，它可以用来描述实体的细节。断言是基于受保

护资源的环境的条件，这些条件必须有效才能够访问资源。相关环境则将被请求的事务处理放置到一个和系统相关的参照系中，该参照系可以基于时间、操作历史或者基于某个规则库中的规则。

(3) 完整性。完整性是指防止某个未授权用户或实体对某个资产进行修改或破坏。它通常与数据完整性同义，数据完整性确保数据未被恶意或偶然修改或破坏。

(4) 可用性。可用性是指保护资产免受可能影响系统有效性的拒绝服务攻击的威胁。从非安全性观点来看（这种观点认为导致系统中断运转的来源是系统本身的缺陷而不是恶意攻击），可用性也是一个关键性的系统属性。但是，针对故障防范、检测、容错、预测和恢复的那些基于软件可靠性理论的策略，却通常在保护系统免受主动且有蓄谋的攻击这方面派不上大用场。在恶意攻击条件下进行可用性建模，和在一般的组件故障条件下进行可用性建模截然不同，因为标准的概率模型不再有效。

(5) 机密性。机密性是指数据对于未授权的用户、实体或过程不可泄露。

(6) 审计。审计是指将所有系统活动已足以再现事件的程度来进行记录。审计通常结合警报同时使用，后者是将触发器与事件进行链接的过程。

(7) 不可否认性。不可否认性是指在某个通信或事务处理过程结束之后，防止其中的任何参与者否认其在该过程中所充当的角色。

安全需求是驱动Web服务安全框架的主要动力。它们在最初的设计和实现阶段中将安全的关注点与Web服务技术的不同领域关联起来。在具体场景中考虑这些需求以获得合适的观点将是非常有用的[10]。

2.2.2 Web服务安全技术

Web服务安全性的保障是通过一系列的安全标准来实现的。这些标准各有特点，互为补充，共同为Web服务提供一个全面、健壮和可伸缩的安全平台。这些安全技术具体总结如表2-1所示[11]。

表 2-1 Web 服务安全技术表
Table 2-1 security technology of Web

技术标准	作用	安全特性
HTTP	验证授权	提供验证功能，限制对保护资源的访问
摘要	完整性	对通信数据进行散列运算，保证完整性
SSL/TSL HTTPS	验证授权、授权、完整性	验证通信双方，加密数据以保证客户与服务器之间的端对端的安全性
XML签名	验证授权、完整性	验证文档完整性和发送者身份，多方可以签署同一文档或者文档的局部
XML加密	机密性、完整性	用公用密钥或私用密钥保护传输数据，多方可以加密同一文档或文档的局部

2.2.3 现有安全技术保护Web服务的不足

目前，Web服务安全采用的是已有的Internet和WWW的一些安全标准，这些标准都是静态的措施，并且其本身和Web服务之间就存在技术上的鸿沟，因此，Web服务的安全问题没有得到根本的解决。Web服务是动态变化的，并且不同应用背景的服务所需要的安全措施可能完全不同。现有的一些安全技术只在一定程度上解决了特定系统的安全问题，在Web服务的集成中并不能发挥令人满意的作用[7]。

Web服务的基本工作过程是通过发送SOAP消息到一个由URL来鉴别的服务点（由一个SOAP Server来接受消息），来请求特定的Web服务（操作），接收到消息的响应结果或者错误提示。在传输层之外，当消息数据被接受和中转的时候，数据的完整性以及其他的安全信息就可能泄露或者丢失。这要求Web服务的请求者或提供者必须信任那些中间节点对消息的获得和处理（那些中间节点可能需要处理消息，生成新的消息）。

除了消息的安全性之外，对于合法的请求方按照消息的内容做出适当的反应和行为，也即权限策略控制都是现有的安全机制无法解决的。现在SOAP通常都绑定在HTTP上进行传送，服务器通常保存一个禁止访问的IP地址列表。这样的安全措施显然是粗糙的，让那些潜在的客户无法访问，Web服务的接口描述文件他们也无法获得，更为全面的安全策略也无法实现。

总之，现有的Web服务安全技术无法完全满足Web服务新的安全需要，因此可将现有的安全技术作为Web服务的第一层保护，同时采用更高层的安全技术来处理Web服务的新的安全问题。

2.3 Web服务安全性规范

2.3.1 Web服务安全主要安全规范

Web服务除了微软和IBM发布的Web服务安全规范WS-Security外，主要有如下的安全规范[17] [18]：

（1）XML加密

XML加密（XML Encryption）是W3C提出的一个规范，它不仅提供了加密XML文档的一部分的方法，而且还提供了加密任何数据和用XML表示加密数据的方法。XML加密使加密功能更容易部署。XML加密语法和处理标准定义了加密数字数据的一个过程和在XML中表示加密的数据的方式。虽然被加密的数据试图比XML更加通用，但是XML数据自然适合于XML加密。

（2）XML签名

XML签名（XML Signature）是由W3C和IETF共同提出的一个规范。它除了解释如何签名XML文档的一部分之外，还解释了如何将任何数据的数字签名表示为XML格式。XML签名技术与其他签名技术的不同之处在于：通过将签名当作原始文档的一部分，XML签名规范消除了在管理和保存数字签名时所需的专用方法，并且提供了一个灵活、标准化的构架，以使多个代理能够签署和保存文档的多个部分。

（3）SAML

SAML（Secure Assertion Markup Language）是OASIS组织提出的一个安全规范。它提供了用XML格式表示有关终端用户的身份验证、授权以及属性信息的方法。SAML不提供身份验证，但是可以表示关于过去发生的身份验证或者授权行为的信息。Web服务要获取长久的成功，其最大的困难在于安全性问题。其中最重要的安全性问题之一就是一些联合系统之间的用户身份验证问题和单节点登录。通过单节点登录，用户可以基于一次身份验证就可以使用多项Web服务或由多个Web服务构成的单个服务，SAML便于人们实现分布式身份验证。

（4）XACML

XACML（XML Access Control Markup Language）是OASIS开发的一个安全规范，它可以用XML格式表示访问控制的规则。SAML断言表示的授权策略可以基于XACML表示的规则之上。当前大多数访问控制和授权系统通常如下实现：处于特定环境中的实体向系统发出请求，系统检索其访问规则或策略，然后对该请求授权或拒绝，这些系统都是专用的、简化的模型。

（5）XKMS

XKMS（XML Key Management Specification）是W3C提出的一项规范，它为Web服务公钥基础设施PKI中密钥的注册和分发而建立的一个规范。通过支持这种Web服务的创建，XKMS能够消除使用PKI的复杂性，使Web服务在设计应用程序时结合安全机制时变得更加容易。应用程序可以简单地将所有PKI处理任务委托给某第三方信任服务，而不是自己来对PKI函数进行编码，这就可以将精力集中在业务逻辑上，同时应用程序将缩小，便于在内存有限的设备上使用。

2.3.2 WS-Security规范

WS-Security主要是一个用于基于XML的安全性元数据容器的规范。它定义了一个具有安全可扩展性的SOAP消息头元素，以这种形式在现有安全标准和规范中添加了一个架构，用于将现有安全机制嵌入到SOAP消息中[4]。但是，WS-Security并不指定其中安全性信息的格式，而是指定如何在SOAP消息中嵌入现有安全机制定义的安全性信息。例如，如果使用XML签名，则可将由XML签名定义的信息包含到SOAP消息头中，其中包括消息的签名方法、使用的密钥以及得出的签名值。同样，如果使用XML加密，则可将加密信息添加到WS-Security消息头中。

将WS-Security与XML数字签名、XML加密、安全令牌等技术结合使用，可提供SOAP消息级的完整性、机密性、用户验证、授权、不可否认性等方面的安全特性。例如，消息完整性是通过同时利用XML签名和安全性令牌来确保消息在传输过程中不被修改而实现的。消息机密性同时利用XML加密和安全性令牌来保持部分SOAP消息是机密的[19]。

微软和IBM在发布了第一个Web服务安全规范WS-Security后，又发布了一系列后续规范计划。如图2-3所示（从下向上的发展方向）。

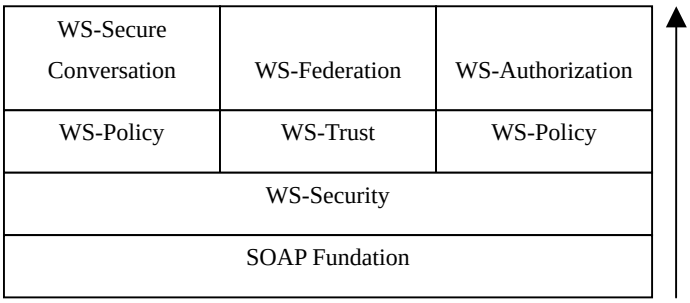

图 2-3 Web 服务安全规范
Figure 2-3 Web services security specifications

这组规范以WS-Security为基础，建立一个初始规范，包括：Web服务端点策略（WS-Policy）、一个信任模型（WS-Trust）和一个隐私权模型（WS-Privacy）。在这个基础上建立了后续规范，包括：安全会话（WS-Secure Conversation）、联合信任（WS-Federation）和授权（WS-Authorization）[20] [21]。

（1）WS-Policy

WS-Policy用于发布Web服务的组织为其Web服务指定安全需要。这些安全需求包括所支持的加密和数字签名的算法、保密性属性以及将这些信息绑定到Web服务上的方式。绑定到Web服务很可能采用WSDL绑定的形式，从而将策略信息附加到Web服务的定义上。

（2）WS-Trust

WS-Trust用于建立直接和代理信任关系（包括第三方和中介体）的模型。模型描述如何通过创建安全性令牌，保证服务把现有的直接信任关系用作代理信任的基础。这些安全性令牌保证服务建立在WS-Security的基础上，用一种保证令牌的完整性和机密性的方式传送那些必需的安全性令牌。然后模型描述如何把几个现有的信任机制与这个信任模型一起使用。最后，这个信任模型将明确允许，但不强求主体进行授权委托和模仿。

（3）WS-Privacy

创建、管理和使用Web服务的组织将经常需要声明它们的隐私权策略，并要求进来的请求声明发送者是否遵守这些策略。通过使用WS-Policy，WS-Security和WS-Trust的组合，组织可以声明并指出遵守声明的隐私权策略。这个规范将描述一个关于如何把隐私权语言嵌入到WS-Policy描述以及如何使用WS-Security把隐私权声明与消息关联起来的模型。最后，这个规范将描述如何使用WS-Trust机制同时为用户首选项和组织实践声明评价这些隐私权声明。

（4）WS-Secure Conversation

WS-Secure Conversation描述Web服务如何认证请求者消息、请求者如何认证服务以及如何互相建立认证的安全性上下文，以及描述如何建立会话密钥、派生密钥和消息令牌密钥。最后，这个规范将描述服务如何安全地交换上下文（关于安全性属性和相关数据的声明的集合）。为完成这个工作，该规范将描述WS-Security和WS-Trust中定义的安全性令牌签发和交换机制概念，并以它们为基础。例如，使用这些机制，服务可以支持使用弱对称密钥技术的安全性令牌，还可以签发使用非共享（不对称）密钥的更强壮些的安全性令牌。WS-Secure Conversation被设计为在SOAP消息层运作，这样消息就可以通过多种传送器和中介体传送。这并不排除在其它消息传递框架中使用它。为进一步增

加系统的安全性，可以跨选中的多个链接把传输级安全性与WS-Security和 WS-Secure Conversation一起使用。

（5）WS-Federation

这个规范定义如何使用WS-Security，WS-Policy，WS-Trust和WS-Secure Conversation规范构建联合信任案例。例如，它将描述如何把Kerberos和PKI基础架构联合起来。同时还引入一个信任策略来指出并限制和确定被代理的信任类型。这个规范还将定义用于管理信任关系的机制。

（6）WS-Authorization

这个规范将描述如何指定和管理Web服务的访问策略。它将特别描述如何在安全性令牌内指定声明，以及这些声明在端点处将如何被解释。该规范与XACML有相互重叠之处[22]。

第三章 风险评估

本章介绍了风险评估的相关概念，风险评估的必要性和风险评估原则；从资产、脆弱点、威胁性三个方面对风险评估流程和算法进行研究分析；最后详细介绍了风险评估的常用方法并对各种方法进行了比较。

3.1 风险评估相关

3.1.1 风险评估基本概念

首先依据国际标准化组织的有关标准对风险评估常见的若干定义做一介绍。风险评估常见的一些基本概念如下[5] [35] [36] [37]：

（1）风险分析（Risk Analysis）：系统地利用信息以识别风险来源和估计风险。

（2）风险评价（Risk Evaluation）：将评估的风险与既定的风险准则进行比较以确定重大风险的过程。

（3）风险评估（Risk Assessment）：风险分析和风险评价的全过程。

（4）信息安全（Information Security）：保护信息的保密性、完整性、可用性及其他属性，如真实性、可核查性、可靠性、防抵赖性等。

（5）资产（Asset）：任何对组织具有价值的东西，包括计算机硬件、通信设施、建筑物、数据库、文档信息、软件、信息服务和人员等，所有这些资产都需要妥善保护。

（6）脆弱性（Vulnerability）：脆弱性是与信息资产有关的弱点或安全隐患。脆弱性本身并不对资产构成危害，但是在一定条件得到满足时，脆弱性会被威胁加以利用来对信息资产造成危害。

（7）威胁性（Threat）：可能导致信息安全事故和组织信息资产损失的活动或能力。威胁是利用脆弱性来造成后果的。

（8）风险（Risk）：指在一定时间内，由于系统行为的不确定性带来危害的可能性。

（9）可能性（Likelihood）：用作对几率或频率的定性描述。

（10）影响（Impact）：或者是后果（Consequence），意外事件发生给组织带来的直接或间接的损失或伤害。

（11）安全措施（Safegard）：也称作控制措施（Control）或对策（Contermeaeure），即通过防范威胁、减少脆弱点、限制意外事件带来影响等途径来减缓风险的机制、方法

和措施。

（12）残留风险（Residual risk）：实施风险处置后仍旧残留的风险。

（13）成本效益分析（Cost-Benefit Analysis）：相对价值与各个提议的控制措施的成本之间的估计与比较，从而实施最有效的控制措施。

3.1.2 风险评估必要性

一个完整的安全体系和安全解决方案是根据网络体系结构和网络安全形势的具体情况来确定的，没有一个“以不变应万变”的通用的安全解决方案。信息安全关心的是保护信息资产免受威胁。

绝对安全是不可能的，只能通过一定的措施把风险降低到一个可接受的程度。对一个信息系统来说，解决信息安全的首要问题就是明白信息系统目前与未来的风险所在，充分评估这些风险可能带来的威胁与影响的程度，做到“对症下药”，这就是信息与网络系统的风险分析与评估的必要性。

风险评估也是当前信息工作客观的需要和紧迫的需求。据了解，在国外对风险评估越来越重视，他们强调没有任何事情比解决错误的问题和错误的系统更有效率了，现在西方国家信息安全领域大大加强了风险评估安全系统的评估工作，通过法规标准加以保证，作为我们国家来说，目前如果没有一个比较统一的评估方法，对全国的安全形势是很难评估全国安全事态的，对一个行业和一个系统如果没有评估方法，很难搞清楚整个行业存在着信息系统问题[23]。

3.1.3 风险评估原则

风险评估在对目标系统存在的漏洞进行分析时应遵从的原则：

（1）没有绝对安全的系统，所有系统都存在安全威胁，可能存在的威胁的数量和种类及其概率都是难以精确预测的。

（2）风险评估过程中以不能对现有的网络资料和网络设备造成损坏为基准，只能采用模拟攻击手段。

（3）风险评估只能根据已知的漏洞及其风险程度进行检测和评估，不能对尚未公布的安全漏洞进行检测和评估。

（4）风险评估人员需要经常关注最新漏洞的公布情况，及时的对漏洞库进行更新。

（5）风险评估必须签署保密协定，严格恪守保密原则，不能以任何形式向外界公布有关被检测对象的内部情况[22]。

3.2 风险评估的流程

3.2.1 资产的识别与评估

资产是企业、机构直接赋予了价值因而需要保护的东西。它可能是以多种形式存在，无形的、有形的、硬件、软件、文档、代码、服务、企业形象等。通常信息资产的机密性、完整性和可用性是公认的能够反映资产安全特性的三个要素。信息资产安全特性的不同也决定了其信息价值的不同，以及存在的弱点、面临的威胁、需要进行的保护和安全控制都各不相同。为此，有必要对企业、机构中的信息资产进行科学识别，以便于进行后期的信息资产抽样、制定风险评估策略、分析安全功能需求等活动。

在一般评估体系中，资产大多属于不同的信息系统，此次首先需要将信息系统及其中的信息资产进行恰当的分类，才能在此基础上进行下一步的风险评估工作。我们将资产分为5个大类和33个小类（参见表4-1），这样可以帮助评估人员更有效地识别组织的关键资产[24]。

在识别了组织的关键资产后就要对这些资产进行赋值，资产赋值是对资产价值的估算，而不是以资产的账面价格来衡量的。资产赋值的过程也就是对资产机密性、完整性和可用性影响分析的过程。影响就是由人为或突发性引起的安全事件对资产破坏的后果。一般情况下，影响主要从以下几方面来考虑：(1）违反了有关法律和规章制度；(2）影响了业务执行；(3）造成了信誉损失；(4）侵犯了个人隐私；(5）造成了人身伤害；（6）对法律实施造成了负面影响；(7）侵犯了商业机密；(8）违反了社会公共准则；（9）造成了经济损失；(10）破坏了业务活动；(11）危害了公共安全。我们列出了11种影响方面并给出了各方面的影响等级的描述（参见表4-2、表4-3)。

在具体实现信息资产赋值的过程中，通常采用半定量的方法，即对每一项评估的固定资产根据其影响按照机密性、完整性、可用性进行半定量赋值。根据ISO13335的定义，机密性是指信息不能被未经认证的实体、主体和处理进程（非法用户）所使用或者披露；（数据）完整性是指数据不会被非法篡改或者损毁；可用性是指资产可以经过认证的实体（合法用户）所使用。对资产机密性、完整性、可用性的赋值采用了半定量的办法。即对资产的机密性、完整性、可用性根据资产的具体情况赋以一个0-4之间的值，代表了对资产机密性、完整性、可用性的要求程度。在此基础上，资产赋值的半定量模型，认为机密性、完整性、可用性能够大体描述信息资产的影响价值，因而可用通过对这三者的逐项赋值从而确定资产的影响价值[25]。亦即：

$$Assets = f(Confidence, Integrity, Available) \quad (3\text{-}1)$$

其中f( )是由资产的机密性、完整性、可用性映射到资产价值的函数，在实际评估中，可以根据实际需要采取多种形式的函数定义。一种实际使用的映射函数形式如下：

$$Assets = Rounds\left\{\log_2\left[\left(2^{机密性}+2^{完整性}+2^{可用性}\right)/3\right]\right\} \quad (3\text{-}2)$$

其中，Round{ }表示四舍五入处理，保留一位小数。

3.2.2 脆弱点的识别与评估

脆弱性识别是风险评估中最重要的一个环节。脆弱性识别可以以资产为核心，针对每一项需要保护的资产，识别可能被威胁利用的弱点，并对脆弱性的严重程度进行评估；也可以从物理、网络、系统、应用等层次进行识别，然后与资产、威胁对应起来。对应用在不同环境中的相同的弱点，其脆弱性严重程度是不同的，评估者应从组织安全策略的角度考虑、判断资产的脆弱性及其严重程度。信息系统所采用的协议、应用流程的完备与否、与其他网络的互联等也应考虑在内。

脆弱性识别主要从技术和管理两个方面进行，技术脆弱性涉及物理层、网络层、应用层等各个层面的安全问题。管理脆弱性又可分为技术管理脆弱性和组织管理脆弱性两方面，前者与具体技术活动相关，后者与管理环境相关。

脆弱性和资产紧密相连，它可能被威胁利用，引起资产损失或伤害。脆弱性本身不会造成损失，它只是一种条件或环境，可能导致被威胁利用而造成资产损失。脆弱性的出现有各种原因，例如可能是软件开发过程中的质量问题，也可能是系统管理员配置不当，或者是管理方面存在隐患。但是，它们的共同特性就是给攻击者提供了机会。

脆弱性的获取可以有多种方式，例如，工具扫描、日志审计、文件完整性检验、病毒检测、渗透测试等。本文重点介绍工具扫描、日志审计、病毒检测等几种脆弱性评估方式。

安全扫描技术基本上可分为基于主机的扫描技术和基于网络的扫描技术两种，前者主要关注软件所在主机上的风险与漏洞，对系统中不合适的设置、脆弱的口令以及其他同安全规则抵触的对象进行检查；而后者则是通过网络远程探测其他主机的安全风险与漏洞，它通过执行一些脚本文件模拟对系统进行攻击的行为并记录系统的反应，从而发现其中的安全漏洞。

工具扫描主要是通过评估工具以本地扫描的方式对评估范围内的系统和网络进行安全扫描，从内网和外网两个角度来查找网络结构、网络设备、服务器主机、数据和用

户账号/口令等安全对象目标存在的安全风险、漏洞和威胁。在信息安全风险评估项目中，工具扫描花费低、效果好、见效快、与网络的运行相对独立、安装运行简单，可以大规模减少评估者的手工劳动，有利于保持全网安全政策的统一和稳定，是进行脆弱性评估的首要方法。扫描器是一种自动检测远程或本地主机安全性弱点的程序，通过使用扫描器你可以不留痕迹的发现远程服务器的各种TCP/IP不同的端口的服务，并记录目标给予的回答，通过这种方法，可以搜集到很多关于目标主机的各种有用的信息[25]。

扫描器并不是一个直接的攻击网络漏洞的程序，它仅仅能帮助我们发现目标机的某些内在的弱点。一个好的扫描器能对它得到的数据进行分析，帮助我们查找目标主机的漏洞。但它不会提供进入一个系统的详细步骤。

可以按照脆弱性存在会造成破坏后果的严重性，采用等级方式对已识别的脆弱性的严重程度进行赋值。不同的等级分别代表脆弱性严重程度的高低。等级数值越大，脆弱性严重程度越高。由于很多弱点反映的是同一方面的问题，或可能造成相似的后果，赋值时应综合考虑这些弱点，以确定这一方面脆弱性的严重程度。本文将脆弱性分为3个等级，分别是高（H）、中（M）、低（L）,并且从高到低分别赋值2-0。

3.2.3 威胁性的识别与评估

威胁是指可能对资产组织造成损害事故的潜在原因。威胁可能源于对企业或组织的信息直接或间接的攻击。威胁也可能源于偶发的、或蓄意的事件。作为风险评估的重要因素，威胁是一个客观存在的实物，无论对于多么安全的信息系统，它都存在。

在这一过程中，首先要对组织需要保护的每一项关键资产进行威胁识别。在威胁识别过程中，应根据资产所处的环境条件和资产以前遭受威胁损害的情况来判断，一项资产可能面临着多个威胁，同样一个威胁可能对不同的资产造成影响。识别出威胁由谁或什么事物引发以及威胁影响的资产是什么，即确认威胁的主体和客体。威胁可能源于意外的，或有预谋的事件。用于威胁评估的信息能够从信息安全管理的有关人员，以及相关的商业过程中获得，这些人可能是人事部的职员、设备策划和IT专家，也包括组织内部负责安全的人员。

威胁具有两个属性：可能性（Likelihood）、影响（Impact），它们可以被赋予一个数值，来表示该属性。本文建议的威胁赋值方法如下[25]：

（1） 可能性赋值参考：可能性的赋值按照该威胁实际发生的概率来估算，具体赋值为1%-10%，即0.01-1；

（2） 影响赋值参考：影响的赋值按照该威胁发生后对资产的实际损失进行估算，

具体赋值为1-100，发生该威胁后，资产完全损失，则赋值为100，其他损失情况进行估算，范围是1-100；

(3) 威胁值=可能性*影响

可能性属性非常难以度量，它依赖于具体的资产、弱点、而影响也依赖于具体资产的价值、分类属性。并且，这两个属性都和时间有关系，也就是说，具体的威胁评估结果会随着时间的变动而需要重新审核。

3.2.4 风险的计算

风险计算就是在对资产、威胁、脆弱点进行完评估后，根据评估的结果计算每一种资产所面临的风险值。根据风险评估过程进行的详细程度可分为结构化的风险计算方式和非结构化的风险计算方式，两种不同的方式可以在实际应用中综合使用。结构化的风险计算方式，对风险所涉及的指标进行详细分析，最终得出风险结果。这种方式通常需要专业人员的参加，风险结论详细。非结构化的风险计算方式通常都是建立在通用的威胁列表和脆弱点列表上，用户可以根据列表提供的线索对资产面临的威胁和威胁可利用的弱点进行选择，从而确定风险。风险计算方式的选择通常要结合使用者主观经验，只要适合组织的需要和实际情况，能产生可信的结论，就可以使用。结构化的风险计算方式有风险矩阵测量、威胁分级法、风险综合评价等，非结构化的风险计算方式有风险矩阵、从员工角度进行风险分析等方法[38]。我们选择的是结构化的风险矩阵测量法，因为这种方法较其他方法更为精确，得出的结论也较为详细（参见表4-7），根据资产、威胁、脆弱点的赋值就可计算出每一项资产的风险值。

3.3 风险评估的常用方法

评估方法在系统风险评估过程中的作用不可忽视[26]，一个评估方法的优劣甚至可以直接影响到评估结果的有效性。国内外现有的风险评估方法很多，这些方法并没有孰优孰劣之分，组织或企业选择风险评估方法的关键是要根据自己的实际情况和要达成的安全目标来进行[27]。目前比较常见的评估方法有三种：定性分析、定量分析、半定量分析（定性分析和定量分析相结合的方法）。

3.3.1 定量评估方法

定量的评估方法运用数量指标对安全风险进行评估。它分析风险发生的概率、风险

危害程度所形成的量化值。典型的定量分析方法有因子分析法、聚类分析法、时序模型、回归模型、等风险图法、决策树法、熵权系数法等。

定量的评估方法优点是用直观的数据来表述评估的结果，看起来一目了然，而且比较客观，定量分析方法的采用，可以使研究结果更科学，更严密，更深刻[27]。缺点是由于分析过程中关联因素较多，而且很多因素难以定量化，例如受攻击概率、“非常严重”与“严重”之间的定量差别等，很难给出精确指标，因此使用该方法进行分析非常复杂而且难以准确[28]。

3.3.2 定性评估方法

定量评估方法常常为了量化，使本来比较复杂的事物简单化、模糊化了，有的风险因素被量化以后还可能被误解和曲解，所以必要的定性分析方法避免了定量方法的缺点，可以挖掘出一些蕴藏很深的思想，使评估的结论更全面、更深刻。

定性的评估方法主要是依据研究者的知识、经验、历史教训、政策走向及特殊变例等非量化资料对系统风险状况做出判断的过程。它主要以与调查对象的深入访谈备案记录为基本资料，然后通过一个理论推导演绎的分析框架，对资料进行编码整理，在此基础上做出调查结论。典型的定性分析方法有因素分析法、逻辑分析法、历史比较法、德尔菲法[27]。这种方法常被用于事务相互作用的研究中，主要解决研究对象“有没有”、“是不是”、“很”、“非常”等问题。

定性分析有两个不同的层次：其一是研究结果本身就是定性的描述材料，没有数量化或者数量水平较低；其二是建立在严格的定量分析基础上的定性分析。很多研究或分析一般都是从研究事物“质”的差别开始，然后研究“量”的规定，在“量”的分析基础上，最后做定性分析，以获得最为可靠的结果。

3.3.3 定量和定性结合评估方法

系统风险评估是一个复杂的过程，需要考虑的因素很多，有些评估要素是可以用量化的形式来表达，而对有些要素的量化又是很困难甚至是不可能的，所以我们不主张在风险评估过程中一味地追求量化，也不认为一切都是量化的风险评估过程是科学、准确的。我们认为定量分析是定性分析的基础和前提，定性分析应建立在定量分析的基础上才能揭示客观事物的内在规律。定性分析则是灵魂，是形成概念、观点，做出判断，得出结论所必须依靠的，在复杂的信息系统风险评估过程中，不能将定性分析和定量分析两种方法简单的割裂开来。而是应该将这两种方法融合起来，采用综合的评估方法[29]。

典型的综合评估方法有层次分析法、故障树法和威胁树法等。

3.3.4 三种评估方法的比较

定量的评估方法直观简单，定性的评估方法全面深刻，定性与定量相结合的评估方法集中了定量评估和定性评估的优点[30]，三种评估方法的具体比较见表3-1。

表 3-1 评估方法的比较
Table 3-1 comparative of assessments

名称	定性评估方法	定量评估方法	定性与定量相结合的方法
定义	主要依据研究者的知识、经验、历史教训、政策走向及特殊变例等非量化资料对系统风险状况做出判断的过程。	运用数量指标来对风险进行评估	定量分析是基础和前提，定性分析是灵魂，是形成概念、观点，做出判断，得出结论所必须依靠的。
举例	因素分析法、逻辑分析法、历史比较法、德尔菲。	因子分析法、聚类分析法、时序模型、回归模型、等风险图法、决策树法等。	失效树分析法、层次分析法等
优点	避免了定量方法的缺点，可以挖掘出一些蕴藏很深的思想，使评估的结论更全面、更深刻。	用直观的数据来表述评估结果，看起来一目了然，而且比较客观，可以使研究结果更科学，更严密，更深刻。	在复杂的信息系统风险评估过程中，将之外两种方法融合起来，取其优点。
缺点	主观性强，对评估者本身的要求很高。	使本来比较复杂的事物简单化、模糊化了，有点风险因素被量化以后还可能被误解或曲解。	难度大，复杂度高。

第四章 Web服务安全风险评估系统总体设计方案

本章在前两章分析介绍Web服务安全和风险评估相关理论的基础上，对Web服务安全风险评估总体方案进行了设计。提出Web服务安全风险评估系统的设计目标、体系结构和评估流程；介绍了Web服务安全风险评估总体方案的数据库结构模型；最后阐述了风险值的算法。

4.1 风险评估系统的设计目标

本评估系统的目标是：有效地评估Web服务系统存在的安全隐患和风险；评估系统所提供的服务及可能存在的安全问题；找出目前的安全策略中存在的一些不足之处，提供一个完整有效的系统安全评估数据库；提供一个完整的Web服务安全风险评估报告，并给出详细的改进措施和建议。Web服务安全风险评估系统的性能主要有：

（1） 使用服务器/浏览器的体系结构。

（2） 系统整合了国内外公认的优秀的自动化扫描工具Nessus，可以完成多项安全工作，如扫描选定范围内的主机的端口开放情况、提供的服务、是否存在安全漏洞等，避免了扫描不完整的情况。

（3） 建立内容丰富的评估信息库，包括资产及资产属性库、威胁行为库包含了大量的调查问卷，涵盖Web服务各类风险要素属性。

（4） 采用问卷调查的方式发现目标系统潜在的安全漏洞。Web服务的很多漏洞是传统的扫描工具扫描不出来的，因此，通过问卷调查的方法分析并确定安全漏洞。

（5） 风险计算采用定性与定量相结合的分析方法给出评估目标系统安全风险等级，使评估结果更加客观、公正。

（6） 提供详细的报告结果，并给出等级参考标准。

因为Web服务安全风险评估系统是一个涉及知识面广、动态性很强的系统，所以除了具有上面的基本功能外，还应该具有以下特性：

（1） 易于配置。因为Web服务安全风险评估系统的目标就是简化管理员的工作，所以应该提供方便友好的用户界面，使得系统易于配置和使用。

（2） 可扩展性是保证一个风险评估系统的质量和长久生命力的关键。首先，

Nessus扫描器是基于插件体制，扩展性强，支持及时的在线升级，可以扫描自定义漏洞或者最新安全漏洞。再者，系统管理员可以对数据库中相应表结构中的记录进行添加、删除、编辑、修改，使用结构查询语句中Delete、Insert、Update等进行操作，从而将不断发现的新的安全漏洞添加进来。

（3） 深度分析。在所有的漏洞检测结束后，应根据检测的结果，对目标系统的潜在的安全问题进行深度的分析，给出分析报告。另外，如果对同一个目标系统进行了周期性的检测，则应对每次的检测结果进行比较，分析安全漏洞的走向和趋势，从而为提高网络安全提供决策。

（4） 漏洞描述应与CVE兼容。由于目前存在多种安全漏洞库，他们对于同样的安全脆弱性往往有着不同的命名和描述方法。为避免这种混乱，就是采用统一的CVE标准的安全漏洞命名方法。

4.2 系统体系结构

为了达到上面的设计目标，经过大量的分析和研究，我们设计了一个基于客户机/浏览器结构的Web服务安全风险评估系统。在客户机上进行测试和评估工作，它主要集成了数据库模块、风险评估模块等几个模块。浏览器提供了方便、友好的用户界面，用于在扫描前配置扫描参数，在测试结束后提交评估报告。图4-1是Web服务安全风险评估体系结构图。

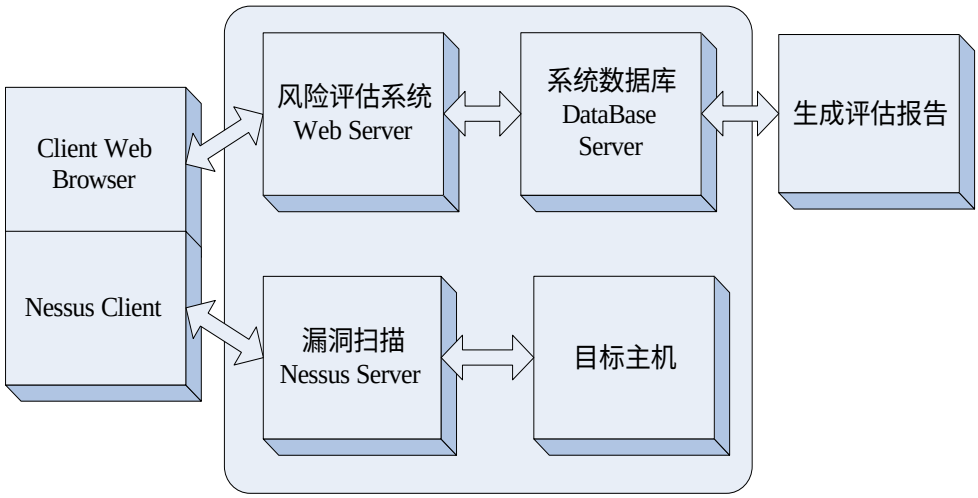

图 4-1 Web 服务安全风险评估体系结构图
Figure 4-1 Web services security
risk assessment system frame

评估者从浏览器端登陆风险评估系统，按照系统提示完成资产识别、脆弱性扫描、威胁性调查三个模块。用户通过进行资产识别完成资产评估模块；通过完成问卷调查来实现威胁性评估模块；脆弱性评估模块则通过用户在Nessus Client设置扫描规则，由系统调用Nessus Server对目标主机进行扫描，然后将扫描结果进行分析后放入数据库。最后评估系统根据各个模块评估结果算出最后的风险值，并给出建议和改进措施。

4.3 系统评估流程的设计

一个评估过程是一个比较系统的过程，因此，需要一个比较合理并行之有效的评估流程的指引。在参考已有风险评估系统的流程设计的基础上，本系统的评估流程设计如图4-2所示。下面简单介绍一下本系统的工作流程：

（1） 建立一个评估任务，由评估发起人进行评估的准备工作。

（2） 评估发起人通过问卷形式依次完成资产评估模块、威胁性评估模块，由系统执行问卷调查分析。

（3） 评估发起人Nessus Client设置扫描规则并完成脆弱性评估模块，由系统执行扫描结果分析。

（4） 综合资产识别、威胁性结果、脆弱性结果进行评估处理。

（5） 最后根据要求生成评估报告。

采用此流程的优点是：①把漏洞扫描的结果作为威胁性的输入，既可以提高准确性，又可以减少评估的复杂性；②通过调查问卷，可以减少定性评估的主观性，提高评估的合理性、准确性和客观性。

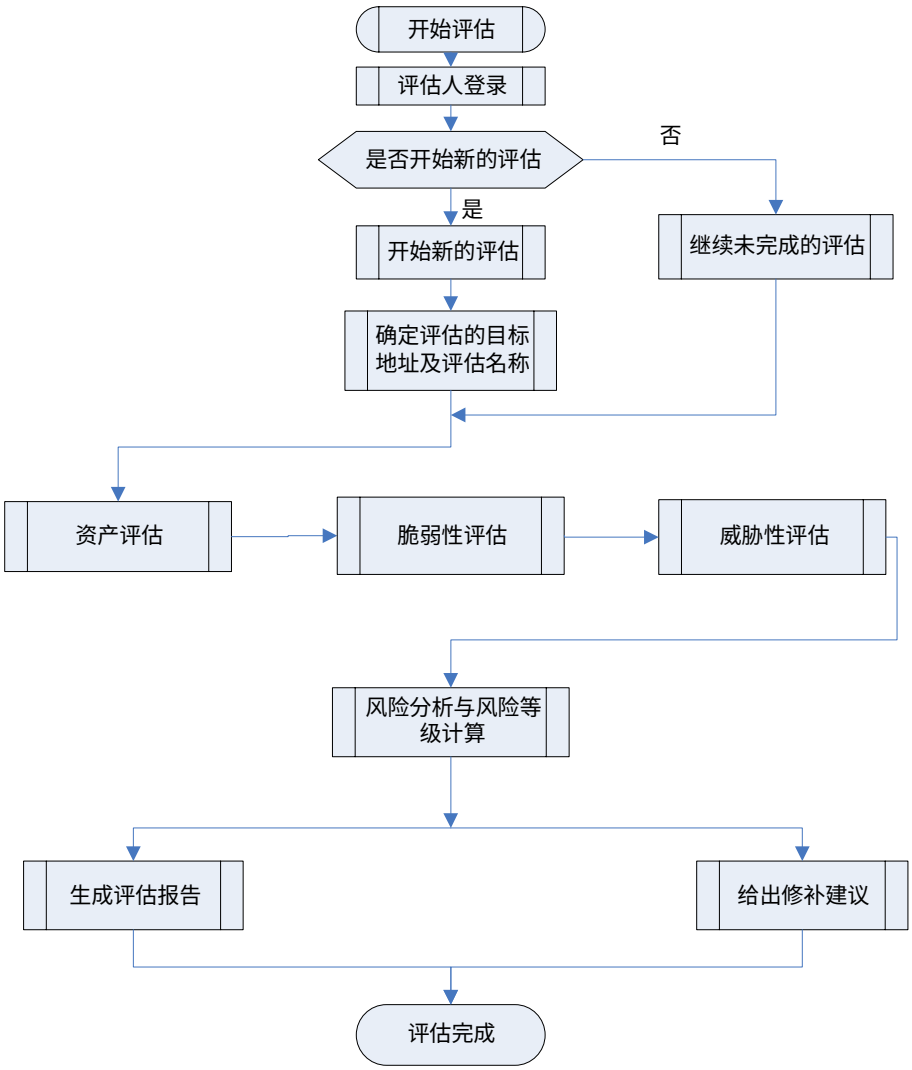

图 4-2 Web 服务安全风险评估流程图
Figure 4-2 Web services security
risk assessment flow chart

4.4 系统数据库的设计及算法

4.4.1 系统建库原则

（1）完整性。数据库是为了存储各种Web服务漏洞的相关信息，包括各种漏洞的信息、漏洞的风险因子的信息等等。所以在建库时力求数据结构的完整性，力求全面地反映Web服务漏洞的特性。

（2）可维护性。安全检测在今后的发展研究中，技术方法会层出不穷。在建库时

要充分考虑数据结构的可扩展性，对于由于某些限制，暂时不能获得的字段信息可保留为空，在获得这些信息后再重新加入，以方便加入新的规则，检测新的漏洞。

（3）权威性。建库时力求全面地分析各种系统漏洞，库中的数据信息来自于官方权威网站，有关编号采用国际统一标准。

4.4.2 数据库结构模型

该数据库设计有11个表，分别是资产分类表 Asset_class、资产影响表 Asset_Impact、资产选项结果表 Asset_Select_Result、评估任务表 Evalu_Task、影响等级表 Impact_level、调查问卷表 Question_table、问卷分类表 Question_ class、问卷选项表 Question_Option、问卷选项结果表 Question_ Select_Result、风险值计算表 Score_Reference、脆弱性扫描结果表 Val _Info。通过对评估任务表的操作可以添加、更改或删除评估任务的相关信息。资产分类表和资产影响表是对资产进行识别的过程，资产选项结果表用来记录评估者评估资产时的选项；调查问卷表、问卷分类表和问卷选项表是通过相关协议和规范来评估 Web 服务安全的威胁性，问卷选项结果表用来记录评估者评估时的选项；脆弱性扫描结果表是记录 Nessus 扫描工具扫描评估对象所得的漏洞表；风险值计算表用来分别记录资产、脆弱性、威胁性的风险值及最终风险等级和最终风险等级的文字描述。数据库结构设计模型如图4-3所示：

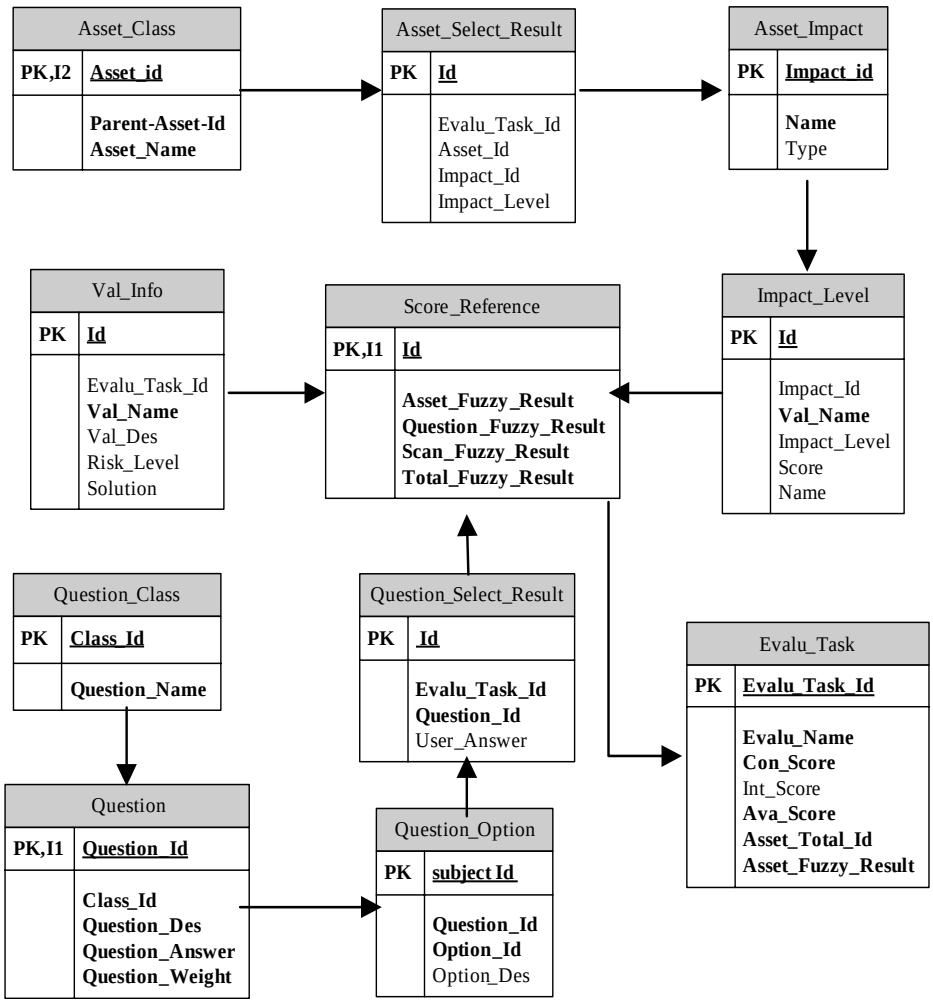

图 4-3 数据库结构设计模型图

Figure 4-3 design model of the database structure

4.4.3 资产评估数据库及算法

资产评估数据库的数据表主要有四个：资产分类表Asset_class、资产影响表Asset_Impact、资产选项结果表Asset_Select_Result和影响等级表Impact_level。

我们将资产从数据、软件、硬件、服务、人员五个方面将资产划分为33个类，如表4-1所示[24]。将资产分类可以帮助用户更好地识别组织关键资产，作为资产评估的基础，识别了组织的关键资产后就要对每一项资产进行赋值，为了帮助用户对资产进行赋值，我们采用了影响表和影响等级表。资产影响表如表4-2所示[24]列出了资产的损失可能造成的后果，共11种，影响等级表对其中的每一种影响都划分了5个等级，如表4-3所示[24]。

表 4-1 资产分类表

Table 4-1 class of asset

数据	软件	硬件	服务	人员
源代码	应用软件	计算机设备	操作系统	业务持有人和股东
数据库	系统软件	网络设备	www、DNS等	高层管理人员
数据文件	开发工具	技术设备	呼叫中心	网络管理人员
系统文档	实用程序	磁介质	内部文件服务	技术人员
运行管理规程		通讯设备	网络连接	中层管理人员
用户手册		机房	网络隔离保护	一般雇员
纸质的各种文件			网络管理	
			网络安全保障	
			入侵监控	
			各种业务生产应用	

表 4-2 资产影响表

Table 4-2 impact of asset

影响编号	描述	属性
1	侵犯经济秘密造成损失。	机密性
2	侵犯商业机密。	机密性
3	违反有关法律和（或）规章制度。	可用性
4	侵犯个人隐私。	机密性
5	对法律实施造成负面影响。	可用性
6	破坏了信誉、声誉。	完整性
7	违反社会公共准则。	可用性
8	危害了公共安全。	可用性
9	侵犯造成了人身伤害。	机密性
10	影响业务执行。	完整性
11	破坏了业务活动。	完整性

表 4-3 影响等级表

Table 4-3 the degree of impact

影响编号	影响等级	影响等级描述
1	1	直接或者间接导致10，000元以下的损失。
1	2	直接或者间接导致10,001-100,000元的损失。
1	3	直接或者间接导致100,001-300,000元的损失。
1	4	直接或者间接导致300,001-1,000,000元的损失。
1	5	直接或者间接导致1,000，000元以上的损失。
2	1	泄露对商业竞争对手有用的信息但是没有商业价值或者商业价值低于10,000。
2	2	泄露对商业竞争对手有用的信息，商业价值在10,001-1,000,000元，并且个人和组织获得不合法收入及好处。
2	3	泄露对商业竞争对手有用的信息，商业价值在1,0 00,001-10,000,000元。
2	4	泄露对商业竞争对手有用的信息，商业价值超过10,000,000元

续　表

2	5	保密或者高度保密信息的泄露可能对国家和商业造成物质损失或者长期的、严重的经济影响。
3	1	触犯法律和规章导致犯罪和民事诉讼造成2,000元以下的损失、惩戒
3	2	触犯法律和规章导致犯罪和民事诉讼造成2,001-10,000元以下的损失、惩戒
3	3	触犯法律和规章导致犯罪和民事诉讼造成10,001-50,000元以下的损失、惩戒或者入狱两年以内
3	4	触犯法律和规章导致犯罪和民事诉讼造成50,001-250,000元以下的损失、惩戒或者入狱两年-十年
3	5	触犯法律和规章导致犯罪和民事诉讼造成50,001-250,000元以下的损失、或者入狱十年以上
4	1	在不违背法律和政策的前提下而侵犯个人信息，造成个人轻微情绪影响
4	2	在不违背法律和政策的前提下而侵犯个人信息，造成个人较大情绪影响
4	3	违背了法律和政策，道德伦理，导致对个人的情绪影响
4	4	违背了法律和政策，道德伦理，造成个人实质性的损失
4	5	违背了法律和政策，道德伦理，造成多人实质性的损失
5	1	促使犯罪的继续发生并且小范围干涉法律的执行
5	2	禁止对罪案的调查和审判
5	3	促使重大罪案的发生并且小范围阻止对重大罪案的调查
5	4	大范围对重大罪案的调查和审判
5	5	采取种种措施完全禁止对重大或者极重大事件的调查和审判
6	1	影响与其他组织的关系
6	2	影响与其他组织和公众的关系，但是公众所指的范围只是限于该组织附近的区域，并且该影响持续时间很短
6	3	影响与其他组织和公众的关系，并且公众所指的范围并不仅仅限于该组织附近的区域
6	4	严重影响与其他组织和公众的关系，并且对广大大众产生影响，但事件不长，对组织的业务影响较小
6	5	严重影响与其他组织和公众的关系，并且对广大大众产生影响，但事件持续非常长，对组织的业务运行造成严重影响
7	1	本地和小区居民的抗议
7	2	导致示威，游行或者本地的工业行动
7	3	出现工业行动，给全国带来影响
7	4	引起全国范围的劳工行为
7	5	严重威胁国家的稳定
8	1	较小
8	2	对组织的有效性和运行造成一定破坏
8	3	对组织的有效性和运行造成较大破坏
8	4	对组织的重要资产的有效性和运行造成较大破坏
8	5	对组织的非常重要资产的持续有效性和运行造成严重破坏
9	1	造成了个人的轻微人身伤害，较短的时间就可以恢复
9	2	造成了个人的较大人身伤害，一个月以内可以恢复
9	3	不但造成身体伤害，也造成了一定的精神伤害，一年以内可以恢复

续　表

9	4	身体和精神都造成了比较大的影响，一年以上的时间恢复，并引起诉讼
9	5	造成了严重的身体伤害和精神伤害，并且很难恢复到以前正常的状态下
10	1	组织的某一部分运转异常
10	2	组织自身的管理以及运行遭到破坏
10	3	破坏组织有效的发展和组织政策的执行
10	4	破坏政府的经济和与其他机构的政策协商
10	5	严重破坏组织重要政策的发展和运行，关闭或者破坏组织的重要操作
11	1	业务活动被破坏了一小部分，关键业务正常，极少的人数就可以修复，经济损失很小，时间少于两天
11	2	有一定的破坏，并且范围较大，但是关键业务运作正常需要一定的人员修复，花费时间低于一个星期，有一定经济损失
11	3	破坏范围和程度较严重，关键业务大部分运作正常，恢复时间低于两个星期，造成经济损失较重
11	4	破坏非常严重，关键业务小部分运作正常，恢复时间低于两个月，经济损失严重
11	5	包括关键业务在内的组织业务大部分或者完全破坏，组织陷于瘫痪状态，需要成立专门的行动小组，管理层进行全局的统筹，共同开展恢复工作，业务完全恢复长于两个月

在识别了组织的关键资产后就要对这些资产进行赋值，评估者通过评估系统对资产的33个小类分别进行11项资产影响的评估，系统根据评估者选择的影响等级对每项资产的每种影响进行相应赋值，然后将各个值按照机密性、完整性、可用性分别进行累计求和，之后分别按照函数(4-1)(4-2)(4-3)将机密性的值、完整性的值、可用性的值转化为三个0-4之间的值，代表对资产机密性、可靠性和完整性的要求程度。机密性、完整性、可用性计算函数分别为：

$$C = \left(\sum_{j=1}^{n} C_j \right) \Big/ (4 * 33) - 1 \qquad (4\text{-}1)$$

$$I = \left(\sum_{j=1}^{n} I_j \right) \Big/ (3 * 33) - 1 \qquad (4\text{-}2)$$

$$A = \left(\sum_{j=1}^{n} A_j \right) \Big/ (4 * 33) - 1 \qquad (4\text{-}3)$$

其中，33代表资产分33个类；C_j为每类资产的机密性，4代表机密性的四个方面，如侵犯商业机密、侵犯个人隐私等，n=132；I_j为每类资产的完整性，3代表完整性的3个方面，如造成业务中断、数据的缺失等，n=99；A_j为每类资产的可用性，4代表可用性的4个方面，如违反有关法律和规章制度、违反社会公共准则、对法律实施造成负面影响等，n=132； 因而可以通过这三者的值确定资产风险级别，一种实际使用的映射函数（4-4）

形式如下：

资产风险级别：

$$Z = Round\left\{\log_2\left[\left(2^C + 2^I + 2^A\right)/3\right]\right\} \tag{4-4}$$

其中，Round{ }表示四舍五入处理。

4.4.4 威胁性评估数据库及算法

威胁性评估数据库的数据表主要有四个：问卷调查表Question、问卷调查分类表Question_Class、问卷调查选项表Question_Option和问卷调查结果表Question_Select_Result。

问卷调查依据参考文献[10]中关于针对Web服务的七个安全性原则划分整个问卷调查表，七个原则包括：认证、授权、完整性、可用性、机密性、审计性和不可否认性。问卷调查根据七个安全性原则从三个不同的角度设置问题。一个是从采用的技术的角度，目前用于加强Web服务安全的技术：SSL/TSL、Kerberos Protocol（Kerberos协议）等；第二个角度是Web服务安全的基础——XML安全和SOAP安全，有以下几个方面：XML签名、XML加密、SOAP消息安全等；第三个角度是从Web服务安全规范的角度，如WS-Security、SAML、XKMS等规范。问卷调查选项表提供问卷问题的可选项。问卷调查结果表用于记录评估者的答案及分值。表4-4是Web服务安全风险评估系统用到的部分调查问卷表及答案。

表 4-4　部分问卷调查表

Table 4-4 part of questionnaire

类别	Web服务安全题目	答案
认证	1.1本Web服务是否提供注册服务机制？	A 是
	1.2本Web服务只提供点到点的认证服务还是提供端到端的认证服务？	B 端到端
	1.8本web服务是否更新现有的身份识别以符合最新Web服务安全规范？	A 是
授权	2.1对访问资源提供大粒度的访问控制还是小粒度的访问控制？	B 小粒度
	2.3 本web服务是否更新现有接入控制安全策略以满足Web服务安全规范？	A 是
	2.4认证成功之后，是否在运行时根据资源访问权限列表来检查服务请求者的访问级别？	A 是
机密性	3.3 本Web服务技术是否能够容纳具有不同密钥长度的各种加密机制？	A 是
	3.2本Web服务是否保证 SOAP消息级的机密性？	A 是
	3.5加密签名数据时，其摘要值是否被加密?(如果没有加密攻击者可以借此推测明文，使得加密数据被破坏）？	A 是
	3.7本Web服务是否保证网络传输层的机密性？	A 是

续　表

完整性	4.3在本Web服务中是否为加密后的数据再采用XML签名以确保初始化矢量的完整性不被破坏？(加密算法中使用的初始化矢量虽然可以解决为给定密钥和数据创建相同密文的安全问题，但初始化矢量本身也可能被修改，使上述问题再次出现。)	A 是
	4.1本Web服务是否采用的多种签名格式？	A 是
可用性	5.2　XML加密的工具对递归深度或请求使用资源数量是否做限制？	A 是
	5.1 请选择本Web服务采用的合适的预防措施以免受任何潜在的拒绝服务的攻击。	A负载平衡、包过滤、备份
审计性	6.2 管理员是否可以在生命周期的不同时刻追踪并找出服务请求？	A 是
	6.5请选择下列哪些技术提供了不可否认性的一个关键元素？	A 数字证书或Kerberos票据的数字签名
不可否认性	7.4 本Web服务技术是否支持不可否认性？（不可否认性使得用户能够证明事务是在拥有合法证书的情况下进行的。）	A 是
	7.5 SOAP消息是否包含时间戳、序列号、有效期、消息相关等元素，并进行签名从而保证消息的唯一性(当缓存这些信息时，可以检测出重放攻击)？	A 是
其它	8.2请选择本web服务受到下列哪些web服务攻击？	E以上攻击都没有发生过
	8.3 是否通过在服务请求上增加简单的特殊信息，本web服务（.asmx）会泄露DISCO或WSDL信息？	B 否
	8.4本Web服务是否安全地交换上下文(关于安全性属性和相关数据的声明的集合)？	A 是
	8.5 是否为本web服务创建诸如.wsdl、 .discomap 、.disco、.xsd 等文件？	E 没有创建任何一种文件
	8.7 本Web服务是否把Kerberos和PKI基础架构联合起来？	A 是
	8.11 将以下哪些保护Web服务通讯的新标准的发展结合到本web服务的安全策略中？	A XML加密 、XML架构、XML数字签名

采用问卷调查的方式来确定Web服务安全的威胁性就是根据评估者对调查问卷做出的选项来识别威胁由谁或什么原因引发以及威胁影响的程度。威胁具有两个属性：被利用的可能性（Likelihood）、造成的影响（Impact），它们可以被赋予一个数值，来表示该属性。本文用到的两个度量的简单标准解释如下[31]：

（1） 影响级别的标准如表4-5：

表 4-5 影响的分级
Table 4-5 the degree of impact

影响等级	等级分值	描述
影响程度低	3	可能造成的损失可以被忽略，对业务基本无损害。
影响程度较低	4	可以造成较小损失并立即可以控制。
影响程度中	5	可以造成损失，业务受到损害、中等财务损失等影响
影响程度较高	6	可以造成重大损失、业务中断、较大财务损失等严重影响。
影响程度高	7	可以造成全部损失或不可用、持续业务中断、巨大财务损失等严重影响

（2）被利用的可能性的标准如表4-6：

表 4-6 被利用的可能性的分级
Table 4-6 the possibility of being used classification

被利用的可能性	被利用的可能性分值	描述
可能性小	0.1	很难被利用。
可能性较小	0.3	费很大的功夫才可以利用。
可能性中	0.5	通过一番周折可以利用。
可能性较大	0.7	通过简单分析便可以利用。
可能性大	0.9	很容易被利用。

威胁性调查的风险级别通过加权求和来实现，映射函数形式如下：

威胁风险级别为：

$$W = Round\left\{0.02 * \left(\sum_{j=1}^{n} P_j * Q_j\right)\right\} \tag{4-5}$$

Round{ }表示四舍五入处理;P为Web服务影响程度、Q为Web服务漏洞被利用的可能性，P_j*Q_j为每个漏洞的风险值,n=50。

4.4.5 脆弱性评估数据库及算法

脆弱性评估数据库的数据表主要有一个：脆弱性扫描结果表Val_Info。该表主要用来记录脆弱性扫描的结果，其字段包括：评估任务编号Evalu_Task_Id、漏洞编号Val_Id、漏洞名称Val_Id、漏洞描述Val_Name、风险级别Risk_Level、解决方案Solution。

脆弱性的获取可以有多种方式，如工具扫描和渗透测试等。工具扫描主要是通过评估工具以本地扫描的方式对评估范围内的系统和网络进行安全扫描，由于其扫描花费低、效果好、见效快、与网络的运行相对独立、安装运行简单，可以大规模减少评估者

的手工劳动，有利于保持全网安全政策的统一和稳定，是进行脆弱性评估的首要方法。本文使用的自动化脆弱性扫描工具是Nessus。

Nessus是一个一体化的弱点扫描工具包，其中包括一系列操作系统和网络服务的许多最新测试。在X-Scan，ISS，Nessus等众多的自动化扫描工具当中，Nessus是最值得称赞的。它基于C/S架构、插件结构的自动化扫描工具，可以免费使用，在线升级并随时获取国内外安全高手编写的最新漏洞的扫描插件。目前Nessus的插件个数已经超过14000个，而且这个数量正在急速上升，因为几乎全世界的安全人员都在使用这个工具，其中有很多黑客会向Nessus提供自己编写的插件。因此，使用Nessus进行扫描就像是全世界的顶尖安全人员都在用他们的技术在帮助我们检查网络中的缺陷[32]。

图4-4是Nessus体系结构图。

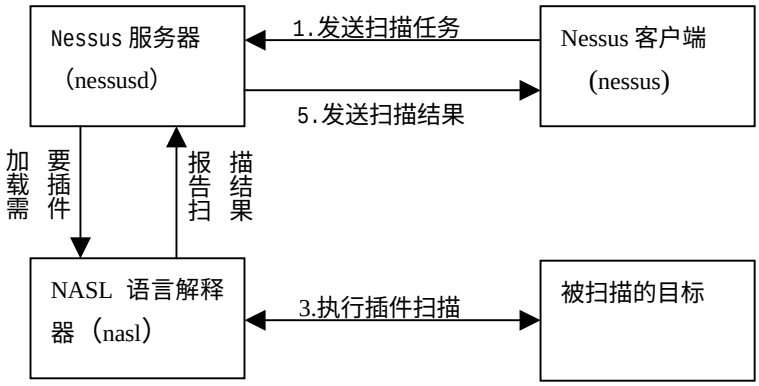

图 4-4 Nessus 体系结构图
Figure 4-4 Nessus system frame

该体系结构体现了一次完整的漏洞扫描过程：(1) 客户端向服务端程序发送详细的扫描任务的参数；(2) 服务端程序接受到客户端程序的请求后，加载完成任务所需要的插件，并合理安排插件的执行顺序；(3) NASL语言解释器执行插件，在执行插件扫描过程中会与扫描目标之间有一些数据交互；(4) NASL解释器判断扫描结果，并报告给服务端程序；(5) 服务端程序归纳从NASL解释器收到的扫描结果，生成漏洞报告反馈给客户端程序。

评估者从浏览器端登陆风险评估系统，在系统提示下调用Nessus扫描器完成脆弱性的扫描，扫描结果保存后上传至风险评估系统，系统将对扫描报告进行分析，并完成脆弱性风险级别的赋值。

Nessus扫描报告使用DOM解析器（Document Object Model，文档对象模型）进行解析。我们使用的是Sun公司的DOM解析器，Sun公司发布的SDK1.5中提供了解析XML文

件的API（Java API for XML Parsing，JAXP），JAXP实现了DOM规范的Java语言绑定，给出了DOM规范指定的接口，并给出实现这些接口的类集合。通过解析获得每个漏洞的ID号，漏洞描述，风险等级和解决方案。

根据Nessus扫描报告分析脆弱性等级时，漏洞对系统的损害程度是评估分析的一个重要指标。对攻击死机型的漏洞，由于利用此漏洞可以使服务器停止服务，甚至导致系统崩溃，则此类漏洞的风险级别在损害程度上是等高的。类似的，攻击入侵型的损害程度是中到高；信息泄露型的损害程度为低到中。因此根据木桶原理[38]：木桶盛水量的最大值取决于木桶中最低木板的高度，即系统受侵害的可能性及程度取决于系统中最危险的漏洞，即风险级别最高的漏洞。本文将脆弱性分为3个等级，分别是高（H）、中（M）、低（L）,并且从高到低分别赋值2-0。

脆弱性级别为：

$$S = \max\{A_1, A_2, A_3, \cdots\}, \forall(A_i),\ \exists(A_i \in [0,2],\ \text{其中} i = 1,2,3,\cdots) \qquad (4\text{-}6)$$

其中，A_i为任何一个漏洞的风险值。

4.4.6 风险评估报告数据库及算法

风险评估报告数据库的数据表主要有两个：风险分值表Score和风险评估报告表Evalu_Task。风险分值表主要用来记录与评估相关的分值，其字段包括：资产评估处理结果Asset_Fuzzy_Result、调查问卷处理结果Question_Fuzzy_Result、脆弱性扫描处理结果Scan_Fuzzy_Result、最终风险处理结果Totall_Fuzzy_Result；风险评估报告表主要用来给评估者浏览最终的Web服务安全风险评估结果，其字段包括：评估任务编号Evalu_Task_Id、评估任务名称Evalu_Task_Name、资产评估机密性的值Con_Score、资产评估完整性的值Int_Score、资产评估可用性的值Ava_Score、资产值Asset_Total_Score、资产模糊化结果Asset_Fuzzy_Result、问卷调查值Question_Total_Score、问卷调查模糊化结果Question_Fuzzy_Result、扫描模糊化结果Scan_Fuzzy_Result、最终风险值结果Total_Fuzzy_Result。

通过风险分值表就可以利用风险矩阵测量法计算最终的Web服务安全风险评估结果了，根据表4-7所列的表格就可判断出风险值。用户在计算完风险值后可根据风险值对风险划分等级，等级划分的范围是根据心理学家提出的“ 人区分信息等级的极限能力为7±2 ”的研究结论，参照文献[33]分为0-8九个等级，如表4-8所示。

表 4-7 风险矩阵
Table 4-7risk matrix

资产值	威胁性等级								
	0			1			2		
	脆弱性等级			脆弱性等级			脆弱性等级		
	低	中	高	低	中	高	低	中	高
0	0	1	2	1	2	3	2	3	4
1	1	2	3	2	3	4	3	4	5
2	2	3	4	3	4	5	4	5	6
3	3	4	5	4	5	6	5	6	7
4	4	5	6	5	6	7	6	7	8

利用求得的资产值、威胁性值、脆弱性值，通过这一表格将风险等级进行确定。例如，如果资产值为3，威胁性为1，脆弱性等级为中，查表可知风险值为5。由上表可以推知，风险矩阵会随着资产值、威胁性、脆弱性等级的增加而扩大。

风险划分等级的表示见表 4-8：

表 4-8 风险划分等级
Table 4-8 the level of risk

等级	描述词	风险等级
0	Absolutely-low	风险度最低
1	Very-low	风险度低
2	Low	风险度偏低
3	Fairly-low	风险度中下
4	Medium	风险度中
5	Fairly-high	风险度中上
6	High	风险度高
7	Very-high	风险度较高
8	Absolutely-high	风险度极高

4.4.7 系统评估算法总结

设计的Web服务安全风险评估算法如下所述：

（1）输入参数：资产评估值Z、威胁性评估值W、脆弱性评估值S；

（2）中间参数：资产机密性评估值C、资产完整性评估值I、资产可用性评估值A、威胁性评估中Web服务影响程度P和Web服务漏洞被利用的可能性Q；

（3）输出参数：风险值R。

风险计算算法分析见图4-5。

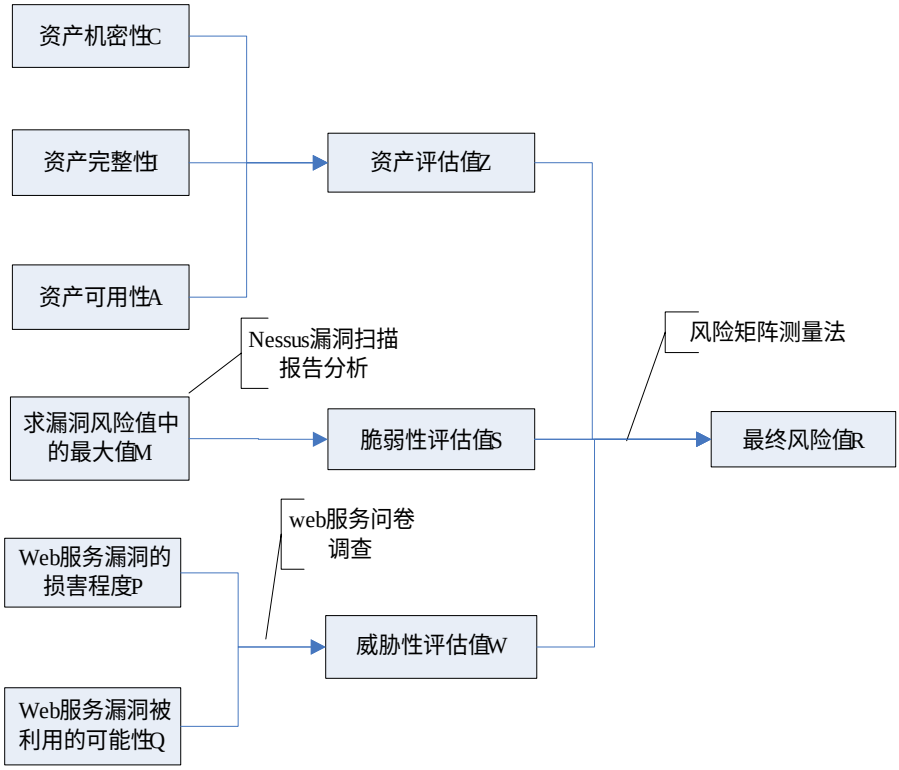

图 4-5 风险计算算法分析图

Figure 4-5 calculation algorithm analysis of risk

计算公式：评估目标系统最终风险值R的参数有三个：资产评估值Z、脆弱性评估值S、威胁性评估值W。风险值R的计算方法查表4-7：

依据风险矩阵测量法，需要首先计算出资产评估值Z，资产评估值Z的求解需先通过资产识别对资产的三个属性：机密性C、完整性I、可用性进行赋值A，然后计算资产评估值。计算公式是：

$$Z = Round\left\{\log_2\left[\left(2^C + 2^I + 2^A\right)/3\right]\right\} \tag{4-4}$$

其次计算威胁性评估值，需要通过对Web服务进行调查问卷分析，利用Web服务漏洞的损害程度P和Web服务漏洞被利用的可能性Q求出威胁性评估值。其计算公式为：

$$W = Round\left\{0.02 * \left(\sum_{j=1}^{n} P_j * Q_j\right)\right\} \tag{4-5}$$

再次计算脆弱性评估值，需要根据对自动化扫描工具Nessus扫描报告进行分析，得出脆弱性评估值。其计算公式为：

$$S = \max\{A_1, A_2, A_3, \cdots\}, \forall(A_i),\ \exists(A_i \in [0,2],\ 其中 i = 1,2,3,\cdots) \qquad (4\text{-}6)$$

上述算法的优点在于：

（1） 每一步所采用的值的值域都是确定的，从而使最终得到的风险值结果也是固定的，风险等级的划分更加准确。

（2） 从风险产生的实际情况，综合考虑了Web服务安全所涉及的主要因素。例如利用扫描工具针对Web服务技术方面的漏洞，利用问卷调查针对Web服务协议规范方面的漏洞等。

（3） 算法实现了定性与定量相结合的分析方法，从而使评估结果更加客观、公正。对资产的识别、安全事件发生的原因以及威胁性调查分析是定性的方法；而对于威胁造成的损害、漏洞被利用的可能性，最终的风险评价则采用定量的方法。

第五章 Web服务安全风险评估系统的实现

根据前面第四章对系统的分析和设计，这一章主要是依据前面的设计结果用软件的形式来实现Web服务安全风险评估系统。下面指出此系统的开发环境和编程工具，并从风险评估系统各个模块详细叙述了其软件实现，最后通过实验分析实验结果并证明Web服务安全风险评估工作的有效性。

5.1 开发环境和工具

从第四章对风险评估系统的结构设计中可以发现，此系统主要由三个模块组成，即数据库模块、风险评估模块和风险评估报告模块。下面简单介绍一下各部分的编程环境和开发工具。

（1） 数据库模块

数据库模块选择Microsoft SQL Server2000作为数据库服务器软件，采用JDBC的数据访问技术，并使用DBCP连接池，对常用的数据库操作方法进行了封装，该数据库模块使用简单、管理方便、可靠性高、力求数据结构的完整，充分地考虑了数据结构的可扩展性。

（2） 风险评估模块

风险评估模块是提供给用户进行操作使用的，所以具有界面的美观性、使用的方便性、编程的便利性等要求，并且该模块的编程不太需要涉及操作系统底层的操作，所以权衡各种编程环境，最终选择了Windows环境，编程语言选择JSP语言进行开发。

（3） 风险评估报告模块

风险评估报告模块提供给用户最终评估结果，评估报告显示评估过程的详细信息包括：评估者、评估时间、IP地址、资产评估值、脆弱性扫描结果、威胁评估问卷调查选择的答案、标准答案、问卷各部分风险值的汇总、整个评估系统的风险等级和风险等级的文字描述。

5.2 风险评估各个模块

本系统风险评估的功能包括：控制整个评估过程；对被评估系统的进行审查和提问；

向漏洞扫描服务器端发送漏洞扫描命令；计算被评估系统的风险等级；生成风险评估报告。

下面按照一个Web服务安全风险评估流程的顺序给出本系统的主界面、评估的申请、资产的识别、脆弱性扫描的获取、威胁性调查问卷的执行、风险的计算以及评估报告的生成等的实现。

5.2.1 数据库模块

数据库技术是信息资源管理最有效的手段。数据库结构设计的好坏将直接对应用系统的效率及实现的效果产生影响。合理的数据库结构设计可以提高数据存储的效率，保证数据的完整和一致。以下是Web服务安全风险评估系统数据库配置语句：

```
private void init() {
        InputStream is = getClass().getResourceAsStream("db.properties");
        dbProps = new Properties();
            try {
                //System.out.println("begin");
                dbProps.load(is);
                //System.out.println("end");
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            //System.out.println("finally");
            driveClassName = dbProps.getProperty("jdbc.driverClassName");
            //System.out.println("driveClassName"+driveClassName);
            url = dbProps.getProperty("jdbc.url");
            username = dbProps.getProperty("jdbc.username");
            password = dbProps.getProperty("jdbc.password");

            initialSize = dbProps.getProperty("dataSource.initialSize");
            minIdle = dbProps.getProperty("dataSource.minIdle");
            maxIdle = dbProps.getProperty("dataSource.maxIdle");
            maxWait = dbProps.getProperty("dataSource.maxWait");
            maxActive = dbProps.getProperty("dataSource.maxActive");
            //  是否在自动回收超时连接的时候打印连接的超时错误
            logAbandoned = (Boolean.valueOf(dbProps.getProperty(
                    "dataSource.logAbandoned", "false"))).booleanValue();

            //  是否自动回收超时连接
            removeAbandoned = (Boolean.valueOf(dbProps.getProperty(
                    "dataSource.removeAbandoned", "false"))).booleanValue();
```

```
// 超时时间(以秒数为单位)
removeAbandonedTimeout = Integer.parseInt(dbProps.getProperty(
        "dataSource.removeAbandonedTimeout", "300"));

//配置DataSource
dataSource = new BasicDataSource();
dataSource.setDriverClassName(driveClassName);
dataSource.setUrl(url);
dataSource.setUsername(username);
dataSource.setPassword(password);

// 初始化连接数
if (initialSize != null)
    dataSource.setInitialSize(Integer.parseInt(initialSize));

// 最小空闲连接
if (minIdle != null)
    dataSource.setMinIdle(Integer.parseInt(minIdle));

// 最大空闲连接
if (maxIdle != null)
    dataSource.setMaxIdle(Integer.parseInt(maxIdle));

// 超时回收时间(以毫秒为单位)
if (maxWait != null)
    dataSource.setMaxWait(Long.parseLong(maxWait));

// 最大连接数
if (maxActive != null) {
    if (!maxActive.trim().equals("0"))
        dataSource.setMaxActive(Integer.parseInt(maxActive));
}

//System.out.println("logAbandoned=" + logAbandoned);
dataSource.setLogAbandoned(logAbandoned);
dataSource.setRemoveAbandoned(removeAbandoned);
dataSource.setRemoveAbandonedTimeout(removeAbandonedTimeout);


} ……
```

5.2.2 风险评估模块

Web服务风险评估模块主要控制风险评估的流程，包括评估者登录、填写评估任务信息、资产评估、威胁性评估、脆弱性评估、风险评估报告生成。评估者正确登陆后可以选择并回答调查问卷并结合漏洞扫描的结果得出风险值、风险等级及其风险等级的文字描述。下面是对风险评估模块中各子模块的介绍。

（1）评估任务管理模块

用户登录之后，进入Web服务安全风险评估系统主界面如图5-1，该界面列出了风险评估的主要步骤，包括创建或选择任务、资产评估、问卷调查、安全扫描、报告生成。主菜单包括评估任务管理、任务信息和系统介绍。在正式评估之前，点击主界面创建进入创建新任务界面如图5-2所示。

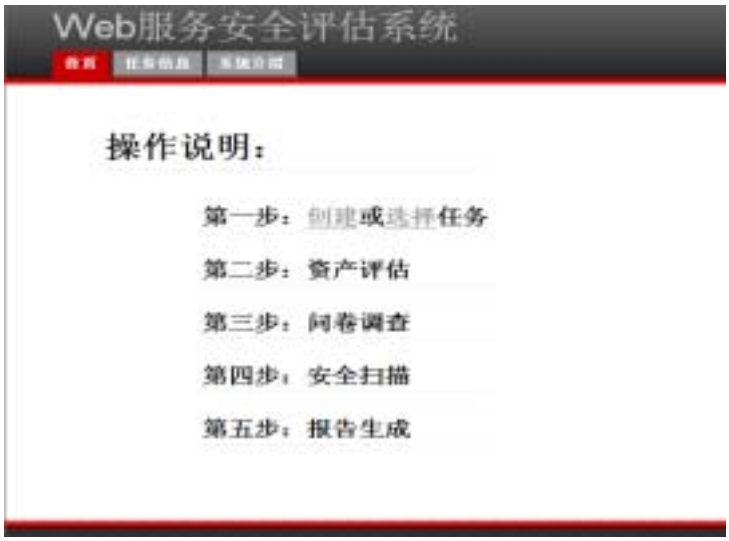

图 5-1 评估主界面

Figure 5-1 assessment main interface

图 5-2 创建新任务界面

Figure 5-2 new task interface

依次填写表单，*为必填项，填写完毕后点击提交按钮将新建的评估任务信息保存到数据库中。同时系统弹出创建成功提示框如图5-3所示。

图 5-3 创建任务提示界面

Figure 5-3 show mission cue interface

若点击主界面的选择任务，则系统进入查看任务界面如图5-4所示,可查看或继续完成以往填写的评估任务。

图 5-4 查看评估任务界面

Figure 5-4 show assessment mission interface

在任务状态一栏中选择需要继续评估的未完成任务，点击设定按钮将未完成的评估任务信息保存到数据库中，继续下一步的评估。

（2）资产评估模块

确定了要评估的任务后就可以正式开始评估，首先要进行的是资产评估。点击主菜单资产评估按钮，将出现如图5-5所示的资产评估界面。

图 5-5 资产评估界面

Figure 5-5 asset assessment interface

资产评估主要是识别资产和对资产赋值。如上图左侧，评估系统的资产评估模块将资产分为5个大类和33个小类，用户可以为资产进行分类从而更好地识别资产，在资产赋值方面，如上图右侧所示，本模块列出了11中影响方面并给出了各方面的影响等级描述，评估者依次完成选项之后将会弹出如图5-6所示的提示框。

图 5-6 资产评估完成界面

Figure 5-6 asset assessment finishment interface

（3）威胁性评估模块

点击确定按钮，则系统进入问卷调查界面如图5-7，问卷调查依据Web服务的认证、授权、完整性、可用性、机密性、审计性和不可否认性七个安全性原则划分整个问卷调查表，问卷调查根据七个安全性原则从三个不同的角度设置问题并提供问卷问题的可选项，评估者的答案将入库进行分析和计算。

图 5-7 威胁性评估界面

Figure 5-7 threaten assessment interface

评估者依次完成选项之后将会弹出如图5-8所示的提示框。

图 5-8　威胁性完成界面

Figure 5-8 threaten finishing nterface

（4）脆弱性评估模块

脆弱性评估模块通过安全漏洞扫描实现，界面如图5-9所示。由于安全漏洞扫描是一个比较复杂的过程，因此需要较高的运行效率和性能要求。比较优秀的漏洞扫描系统大部分采用的是C/S模式，其中服务器端负责漏洞的扫描操作，客户端一般只是起到控制作用。本系统为了减少对已有产品的重复开发，直接使用现有的软件——当前最受欢迎的安全漏洞扫描工具Nessus，并且此软件又是一个免费产品，因此，它完全合乎此系统对漏洞信息获取的要求。此漏洞服务器端运行Nessus Server，其运行环境是Linux操作系统，而Nessus Client 是运行在评估系统上[40]。

图 5-9　脆弱性评估界面

Figure 5-9 frangibility assessment interface

若评估系统所在主机上未安装Nessus安全漏洞扫描软件，则点击下载进入Nessus的官方网站http://www.nessus.org/download/下载，之后按照界面所示步骤完成客户端的安装、扫描，最后将扫描结果上传。Nessus的下载界面如图5-10所示，扫描结果如图5-11所示。

图 5-10　nessus 下载界面

Figure 5-10 nessus download interface

图 5-11　nessus 扫描结果界面

Figure 5-11 nessus report interface

评估者将扫描结果保存并上传后弹出如图5-12所示提示框。

图 5-12　脆弱性评估完成界面

Figure 5-12 frangibility finishing interface

（5）风险分析与计算模块

脆弱性安全扫描结束后，点击子菜单确定进入风险计算界面如图5-13所示。

图 5-13 风险计算界面

Figure 5-13 risk account interface

分别点击计算资产评估按钮、计算问卷调查按钮、计算威胁等级按钮、计算总结果按钮将完成各风险值的分析与计算（见图5-14、图5-15、图5-16、图5-17）。

图 5-14 资产计算界面

Figure 5-14 asset account interface

下面是资产风险值算法的部分代码：

```
public EvaluTask computTotalScore(int evaluTaskId) {
    // TODO Auto-generated method stub
    String sql = "select type ,sum(b.Score) Total_Score from Asset_Select_Result a, Impact_Level b,Asset_Impact c"+
        " where a.Impact_Id = b.Impact_Id and  a.Impact_Level = b.Impact_Level and a.Impact_Id = c.Impact_Id and  Evalu_Task_Id="+evaluTaskId+
        " group by c.type order by type";
```

```
log.debug(sql);
DbTrans db = new DbTrans();
ResultSet rs = null;
int totalScore = 0;
EvaluTask evaluTask = new EvaluTask();
try {
    rs = db.executeQuery(sql);
    int resultCount = 0;
    while(rs.next()){
        resultCount++;
        int scoreFirst = rs.getInt("Total_Score");
        log.debug("scoreFirst"+scoreFirst);
        int score = 0;
        if(rs.getInt("type")==Constants.IMPACT_AVA){
            score = scoreFirst/Constants.IMPACT_AVA_BASE_SCORE-1;
            log.debug("ava:"+score);
            evaluTask.setAvaScore(score);
        }else if(rs.getInt("type")==Constants.IMPACT_CON){
            score = scoreFirst/Constants.IMPACT_CON_BASE_SCORE-1;
            log.debug("con:"+score);
            evaluTask.setConScore(score);
        }else if(rs.getInt("type")==Constants.IMPACT_INT){
            score = scoreFirst/Constants.IMPACT_INT_BASE_SCORE-1;
            log.debug("int:"+score);
            evaluTask.setIntScore(score);
        }
        totalScore +=score;
    }
    int fuzzyScore = 0;
    if(resultCount==3){
        fuzzyScore = this.getAssetFuzzyScore(evaluTask);
```

```
                evaluTask.setAssetFuzzyResult(fuzzyScore);
                evaluTask.setAssetTotalScore(totalScore);
            }else{
                evaluTask.setAssetTotalScore(-1);
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } finally{
            db.close();
            this.closeRS(rs);
        }
        return evaluTask;
    }
```

com.tyut.service.AssetSelectResultServic

```
    private int getAssetFuzzyScore(EvaluTask evaluTask) {
        // TODO Auto-generated method stub
        double twod = 2.0;
        double avad = Math.pow(twod, evaluTask.getAvaScore());
        double cond = Math.pow(twod, evaluTask.getConScore());
        double intd = Math.pow(twod, evaluTask.getIntScore());
        double aved = (avad+cond+intd)/3.0;
        double resd = MathUtils.log(twod, aved);
        return MathUtils.getRound(resd);
    }
```

图 5-15 威胁性计算界面

Figure 5-15 threaten account interface

下面是威胁性风险值算法的部分代码：

com.tyut.service.QuestionSelectResultService

```
/**
    * 计算评估任务的问卷调查总分和模糊化结果
    * @param evaluTaskId
    * @return
    */
   public EvaluTask computTotalScore(int evaluTaskId) {
        // TODO Auto-generated method stub
        //得分
        float totalScore = getQuestionTotalScoreByEvaluTaskId(evaluTaskId);
        EvaluTask evaluTask = new EvaluTask();
        evaluTask.setQuestionTotalScore(totalScore);
        //模糊化
        int fuzzyScore = MathUtils.getRound(totalScore*0.02);
        evaluTask.setQuestionFuzzyResult(fuzzyScore);
        return evaluTask;
   }
```

图 5-16 脆弱性计算界面

Figure 5-16 frangibility account interface

下面是脆弱性风险值算法的部分代码：

```
com.tyut.service.AnalysisReportService
                //依据Risk_Factor确定Risk_Level的值;
                if (Risk_Factor.indexOf("None") != -1) //存在none
                    db_Risk_Level = 0;
                else if (Risk_Factor.indexOf("Medium") != -1)
                    db_Risk_Level = 1;
                else if (Risk_Factor.indexOf("High")!= -1)
                    db_Risk_Level = 2;
                else
                    db_Risk_Level = 0;

                db_Solution = Solution;

                //测试代码
                System.out.println("i = " + i);
                System.out.println("任务编号:" + db_Evalu_Task_Id);
                System.out.println("漏洞编号:" + db_val_Id);
                System.out.println("漏洞名称:" + db_Val_Name);
                System.out.println("漏洞概要:" + db_Val_Synopsis);
                System.out.println("漏洞描述:" + db_Val_Des);
                System.out.println("风险级别:" + db_Risk_Level + "--->" + Risk_Factor);
                System.out.println("解决方案:" + db_Solution);
                rec = new
ValInfo(db_Evalu_Task_Id,db_val_Id,db_Val_Name,db_Val_Synopsis,db_Val_Des,db_Risk_
Level,db_Solution);
                list.add(rec);
                //缺少:入库操作
            }

            //不同ReportItem之间的分隔符
            System.out.println("==================================");
        }
        return list;
    }……
```

图 5-17 最终风险值计算界面

Figure 5-17 risk account interface

下面是最终风险值算法的部分代码：

com.tyut.service.ScoreReferenceService

```
public int    getFinalScoreByEvaluTask(EvaluTask task)
{
        int finalScore=-1;
        String sql = "select * from Score_Reference where Asset_Fuzzy_Result="+task.getAssetFuzzyResult()
            +" and Question_Fuzzy_Result="+task.getQuestionFuzzyResult()
            +" and Scan_Fuzzy_Result="+task.getScanFuzzyResult();
        log.debug(sql);
        DbTrans db = new DbTrans();
        ResultSet rs = null;
        try {
            rs = db.executeQuery(sql);
            if(rs.next())
            {
                finalScore = rs.getInt("Total_Fuzzy_Result");
            }
        } catch (SQLException e)
        {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }finally
       {
            this.closeRS(rs);
            db.close();
        }
        return finalScore;
    }
```

5.2.3 风险评估报告模块

利用风险评估模块各个子模块的评估结果，就可以对整个Web服务系统进行安全风险评估了。风险评估报告模块提供了评估过程的详细信息：评估综述、资产评估相关分值、威胁性调查问卷的分值和脆弱性漏洞扫描汇总、整个评估系统的风险等级和风险等级的文字描述。

生成评估报告包括的内容如图5-18所示：

1、综述
◇1.1 系统名称：
◇1.2 评估者：
◇1.3 评估时间：
◇1.4 IP 地址：
◇1.5 任务总评：
◇1.6 备注：
2、资产评估
◇2.1 机密性风险值（0-4）：
◇2.2 完整性风险值（0-4）：
◇2.3 可用性风险值（0-4）：
◇2.4 资产评估最终结果（0-4）：
3、威胁性评估
◇3.1 问卷调查最终结果（0-2）：
◇3.2 问卷问题及用户的答案和标准答案
◇3.3 八个部分分别的风险值和总的风险值
4、脆弱性评估
◇4.1nessus 扫描的信息：
◇4.2 风险的最高级别：
◇4.3 漏洞信息表及修复措施：
5、Web 服务安全风险评估最终结果
◇5.1 评估系统总的风险值（0-8）：
◇5.2 饼状图显示：
6、参考标准（链接）
◇6.1 风险计算结果参考
◇6.2 威胁性风险评估参考

图 5-18 生成评估报告内容
Figure 5-18 the content of evaluation report generation

5.3 实验

实验应用Web服务安全风险评估系统对选定的目标主机进行了安全风险等级的评估。通过测试主机对目标主机进行风险评估，得出运行在目标主机的目标系统的Web服务安全风险等级。下面是本实验的测试环境、测试用例和测试结论分析。

5.3.1 测试环境

我们的测试实验基于下面的环境：选择校园网上的两台PC机作为实验环境，一个作为目标主机，在其上运行有评估目标系统，目标主机IP地址是219.226.108.206；一个

作为测试主机，其上安装有Web服务安全风险评估系统和安全漏洞扫描工具Nessus3.0。

（1）目标主机和测试主机的硬件配置如下：

CPU：Intel（R） Core(TM)2 1.80GHz；内存：1.00GB；

（2）软件环境：①测试主机的操作系统是Windows xp；②测试主机的编译环境是JDK1.6.0；③应用服务器是Tomcat6.0；④开发环境是MyEclipse 5.5。

5.3.2 测试用例

（1）填写评估信息

点击图标启动Web服务安全风险评估系统，评估之前先填写评估任务名称、评估人、服务器IP等内容。如图5-19所示。

图 5-19 评估信息界面

Figure 5-19 assessment information interface

（2）进行资产评估

点击Web服务安全风险评估系统菜单中资产评估按钮，依次完成资产识别，最终资产评估结果如图5-20所示。

图 5-20 资产评估值界面

Figure 5-20 asset assessment result interface

（3）进行威胁性评估

点击Web服务安全风险评估系统菜单中威胁性评估按钮，依次完成Web服务漏洞相关的问卷，最终威胁性评估结果如图5-21所示。点击图中问卷问题及用户的答案和标准答案可直观用户答案和标准答案，当用户选择的答案与标准答案不一致时，用户可以参照标准答案所表示的安全要求对评估系统进行改进，以满足Web服务安全要求。

3.威胁性评估

◇3.1 问卷调查最终结果（0-2）： 1
◇3.2.1 问卷问题及用户的答案和标准答案
◇3.2.2 八个部分分别的风险值和总的风险值（107.5）

类别名称	总分值	您的分值
认证	21.8	12.5
授权	7.6	2.1
机密性	14.1	11.7
完整性	5.2	1.8
可用性	9.2	5.0
审计和不可否认性	11.4	3.8
SOAP消息安全和安全令牌	14.8	8.5
其它	23.7	13.0
总分值	107.5	58.4

图 5-21 威胁性评估值界面
Figure 5-21 threaten assessment result

（4）进行脆弱性评估

点击Web服务安全风险评估系统菜单中脆弱性评估按钮，调用Nessus扫描安全漏洞，将扫描结果分析入库后得到脆弱性评估结果如图5-22所示。

4.脆弱性评估

◇4.1nessus扫描的信息： 219.226.108.206
◇4.2风险的量化级别： Medium
◇4.3漏洞信息表及修复措施

编号	漏洞编号	漏洞名称	漏洞概要	漏洞描述	风险级别	解决方案
0	22964	blackjack (1025/tcp)Service detection	No Synopsis	No Description	0	No solution
1	22964	unknown (6059/tcp)Service detection	No Synopsis	No Description	0	No solution
2	10622	pptp (1723/tcp)PPTP Detection	A VPN server is listening on the remote port	The remote host is running a PPTP (Point-to-Point Tunneling Protocol) server. It allows users to set up a tunnel between their host and the network the remote host is attached to.	0	Disable this software if you do not use it

图 5-22 脆弱性评估值界面
Figure 5-22 frangibility result interface

（5）计算风险值

完成资产评估、威胁性评估和脆弱性评估后分别计算其风险值并算出Web服务安全最终风险值，如图5-23所示。

图 5-23 风险值计算显示界面

Figure 5-23 risk value account display interface

（6）生成评估报告

点击报告生成按钮，下面图5-24显示了部分界面

图 5-24 生成评估报告界面 1

Figure 5-24 evaluation report generation interface 1

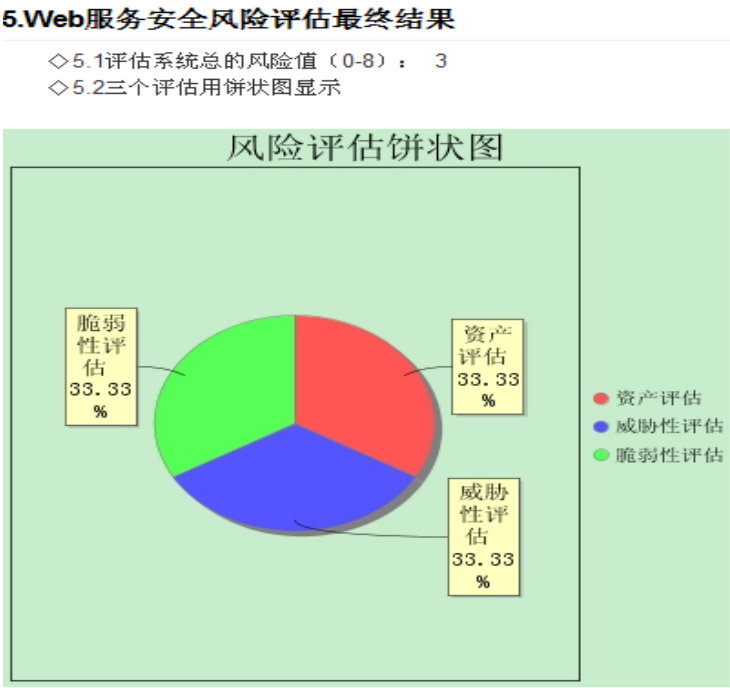

图 5-25 生成评估报告界面 2

Figure 5-25 evaluation report generation interface 2

图 5-26 生成评估报告界面 3

Figure 5-26 evaluation report generation interface 3

从上面生成的评估报告中可直观的看出系统的Web服务安全风险值为3，点击上图中风险计算结果参考，从弹出的文档中可以看到风险等级的文字描述是Fairly-Low（风险度中下），属于九级中的第四级，因此该系统的Web服务安全问题不太严重。同时可以看出被测系统的资产、脆弱性和威胁性均存在需要改进的地方。

5.3.3 测试结论分析

测试结果表明，Web服务安全风险评估系统实现了其基本功能，给出测试目标系统的风险等级，基本上实现了预期的设计目标，证明Web服务安全风险评估系统的可行性。它具有如下优点：

第一，Web服务安全风险评估系统通过对评估目标系统进行风险评估，生成了风险

评估报告，得出了评估目标系统Web服务安全的风险等级，并帮助客户清楚Web服务潜在的风险所在，有助于提高评估目标系统的Web服务安全防护能力。

第二，建立了内容丰富的评估信息库，包括资产及资产影响库、脆弱点库，威胁性调查问卷库，涵盖多类风险要素属性。

第三，对综合工具Nessus扫描结果进行进一步的分析和模糊处理，客观反映了系统的安全状态。

第四，风险计算采用定性与定量相结合的分析方法，使评估结果更加客观、公正。

综合Web服务安全风险评估系统的各项性能，我们认为Web服务安全风险评估系统也存在一些问题。在以后的进一步工作中，可以从以下几个方面对其进行完善。

（1）风险级别的确定问题

由于没有现成的Web服务漏洞参数的度量标准可以参考，用于Web服务安全风险评估系统的Web服务漏洞参数的度量标准是参考众多资料分析总结出来的，还有待继续完善。

（2）风险评估是一个非常复杂的过程，尽管评估方法日益成熟，仍不可避免地会涉及到种种不确定性，包括客观信息的不完整性，如因条件所限未能获得足够的资料或数据等。下一步应该采用多种评估方法不断完善Web服务安全风险评估系统。

（3）Web服务安全风险评估系统的脆弱性评估重点考虑安全工具扫描，没有将日志审计、病毒检测、渗透测试考虑进来。本人将在未来的工作中继续进行这些问题的探索与研究，以求取得更多更好的研究成果。

第六章 总结和展望

目前Web服务技术炙手可热，可是Web服务的安全领域却是有待开发的大西部。Web服务安全问题越来越引起人们的重视，有很多问题是常规的网络安全技术和安全产品无法解决的，而风险评估为确定Web服务潜在的脆弱性和威胁性提供了一种重要方法。通过风险评估可以了解Web服务目前与未来的风险所在，从而为安全策略的确定，Web服务安全运行提供了依据。可以说风险评估技术在Web服务安全技术领域占据重要的地位，对它的研究与开发已经成为必然趋势。

首先，论文从Web服务的发展趋势出发，介绍了Web服务安全技术和规范，分析了Web服务安全现状以及对Web服务进行安全风险评估的意义。第二，在对国内外信息安全风险评估标准研究的基础上，提出了Web服务评估流程和算法并编程实现了该评估系统。第三，目前风险评估工具主要集中在技术方面，而我们设计的系统则针对Web服务的安全技术和安全规范，建立了内容丰富的评估信息库，包括资产及资产影响库、威胁性问卷调查库等，涵盖多类风险要素属性，使评估结果更准确客观。第四，对漏洞扫描Nessus的扫描结果进行进一步的入库分析，客观反应系统的安全状态。

虽然本文对Web服务安全风险评估进行了广泛的研究，设计了一个切实可行的评估系统，但仍存在一些待完善的后续工作，下一步的工作研究重点应着眼于以下三个方面：

第一，论文只研究了系统风险分析与量化评估部分，还需要进一步将安全需求转化为可实现或可执行的安全措施，实现系统风险的自适应控制。

第二，本系统资产信息和漏洞信息的信息量还不足，需要不断充实完善，下一步将向着更加全面、更加动态的模型方向发展，使其结果的可信度和客观性都会有很大提高。

第三，本系统的设计已经涉及到了结合安全技术检测工具，下一步的工作，是希望能够与诸多安全技术的检测工具相结合，为用户和组织提供更为全面细致的安全管理建议。

随着Web服务安全风险评估研究的深入，评估系统中的信息库内容必将得到不断更新，用户可以不断更新数据库来完成这点，使评估系统不断升级。Web服务安全风险评估研究是一项长期性的工作，还需要通过不断的努力探索理论、完善Web服务安全风险评估系统。限于作者水平有限，文中难免有错误和不当之处，恳请各位老师和同学批评、指正。

参考文献

[1] 岳昆，王晓玲，周傲英．Web服务核心支撑技术研究综述［J］．软件学报，2004，15(3)：428-442．

[2] He, F, Le, JJ.Apply the technology of RBAC and WS-Security for secure Web services environment in campus. PROCEEDINGS Of 2006 INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, VOLS 1-7 : 4406-4411 200．

[3] 李林广．Web服务安全技术研究与应用［D］．南京：河海大学，2005．

[4] Lin, MQ, Wang, QM, Li, JH.Methodology of quantitative risk assessment for information system security.LECT NOTES ARTIF INT 3802: 526-531 2005．

[5] 王英梅，王胜开等．信息安全风险评估［M］．北京：电子工业出版社，2007：2-3．

[6] 张振兴．Web服务安全性的研究与实现［D］．河北：华北电力大学，2003．

[7] 国内外信息安全风险评估发展概要，http://www.isra.infosec.org.cn/fxpg/fxpgln/200611/6516.html，2006．

[8] 张法权，张昭时，王泽兵等．Web服务的安全性［J］．计算机时代，2003，4： 1-3．

[9] 于普漪．Web服务安全性研究与实践［D］．武汉：中国地质大学，2007．

[10] 钟鸣．Web服务安全技术的研究与实现［D］．长沙：国防科学技术大学，2004．

[11] 杨剑．Web服务综合安全模型及若干关键技术研究［D］．武汉：华中科技大学，2004．

[12] Feisi Science and Technique Center of Research and Development. Application and Development of Java Web Services. Beijing: Publishing House of Electronics Industry, 2002. pp:190-196

[13] Shmueli O. Architectures for internal Web services deployment. In: Apers P, ed. Proc. of the 27th Int'1 Conf. on Very Large Data Bases. Roma: Morgan Kaufmann Publishers, 2001, pp: 641-644

[14] W3C Web Services Activity. http://www.w3.org/2002/ws/．

[15] OASIS. http//www.uddi.org/whitepapers.html.2003．

[16] W3C SOAP Version 1.2 part 1 Messaging Framework[EB/OL] W3C Candidate Recommendation. http// www.w3.org/soapl2-part1. 2002．

[17] 杨青，怀进鹏，徐枞巍．基于SAML的协同电子商务安全服务系统［J］．计算机

工程与应用，2002，38(14)：228-231．

[18] He, F, Le, JJ, Apply the technology of RBAC and WS-Security for secure Web services environment in campus, PROCEEDINGS OF 2006 INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 2006, VOLS 1-7, pp: 4406-4411

[19] 韩涛，郭荷清．Web服务安全模型的研究与实现［J］．计算机工程，2006，32(10)：130-134．

[20] IBM Corp, IBM Security in a Web Services World, A Proposed Architecture and Roadmap,http://www-900.ibm.com/developerWords/cn/webservices/ws-secmap/indexeng.shtml, 2004．

[21] 王继梅，金连甫．Web服务安全问题研究和解决［J］．计算机应用与软件，2004，21(2)：91-93．

[22] 武晓娟． Web服务安全风险评估［D］．太原：太原理工大学，2008．

[23] 赵冬梅，马建峰，王跃生．信息系统的模糊风险评估模型［J］．通信学报，2007，28(4)：51-55．

[24] 崇小蕾．信息安全风险自评估方法的研究及其辅助工具的设计与实现［D］．西安：西安电子科技大学，2006．

[25] 许黎．基于漏洞检测的网络安全风险评估系统的研究与实现［D］．成都：电子科技大学，2007．

[26] 郭红芳，曾向阳．风险分析方法研究［J］．计算机工程，2001，27(3)：131-132．

[27] 冯妍．网络安全风险评估系统的研究与设计［D］．西安：西北大学，2006．

[28] 上海市信息安全测评认证中心网站，http://www.shtec.gov.cn.

[29] 周明．信息系统风险评估的计算机辅助系统设计［D］．郑州：解放军信息工程大学，2005．

[30] 韩权印．基于BS7799的信息安全风险评估研究与设计［D］．西安：西安电子科技大学，2005．

[31] 邹湘河．漏洞检测与风险评估技术研究［D］．成都：电子科技大学，2005．

[32] 国家反计算机入侵和防病毒研究中心．网络安全攻防实战［M］．北京：电子工业出版社，208：84-86．

[33] 唐慧林．基于模糊处理的系统风险评估方法研究［D］．郑州：中国人民解放军信息工程大学电子技术学院，2005 ．

[34] PingWang, Kuo-Ming Chao,Chi-Chun Lo,Chun-Lung Huang,Muhammad Younas,A

fuzzy outranking approach in risk analysis of Web service security, Springer Science+Business Media, 2007

[35] 仲维国．信息安全风险评估和管理工具研究．http://www.infosec.org.cn/files/paper2004/tool.doc，2004．

[36] 微软安全风险管理指南V1.0．http://www.microsoft.com/china/technet/securiy/guidance/secrisk/default.mspx,2006．

[37] 孙仁科，肖永全，张玉东．Web服务的安全机制和安全模型［J］．计算机应用与软件，2005，22(6)：112-114．

[38] 孙鹏鹏．信息安全风险评估系统的研究与开发［D］．北京：北京交通大学，2007．

[39] 上海市信息安全测评认证中心网站．http://www.shtec.gov.cn.

[40] Nessus的官方网站．http://www.nessus.org/download/．