CN43-1258/TP
ISSN 1007-130X
计算机工程与科学
COMPUTER ENGINEERING & SCIENCE
2007 年第 29 卷第 10 期
Vol. 29, No. 10, 2007

文章编号:1007-130X(2007)10-0011-03

Web 服务安全性测试技术研究*
Research on the Web Services Security Testing Technology

施寅生,邓世伟,谷天阳
SHI Yin-sheng, DENG Shi-wei, GU Tian-yang
(北京系统工程研究所,北京 100101)
(Beijing System Engineering Institute, Beijing 100101, China)

摘　要:Web 服务的应用越来越广泛,Web 服务中的安全缺陷与漏洞也在不断增多,Web 服务安全性问题日益突出。Web 服务安全性测试是保证 Web 服务软件安全性、降低安全风险的重要手段。本文提出了一种 Web 服务安全性测试框架,论述了 Web 服务主要的安全功能需求、实现标准及实施安全功能测试的一般原理,并从攻击 Web 服务的角度对 Web 服务安全漏洞测试进行了系统介绍,分析了 Web 服务常见的安全漏洞及测试方法。

Abstract: Web services are applied more and more widely . The security flaws and vulnerabilities in Web services are growing. Web services security have become increasingly prominent. Web services security testing is an important means to ensure Web services security and decrease security risks. This paper presents a Web services security testing framework, and investigates the main security function requirements and implementation standards of Web services. It also discusses the principle of implementing security function testing. From the perspective of Web services attacking, it discusses the Web services security vulnerability testing, and analyzes the test methods for the common vulnerabilities.

关键词:Web 服务;安全性测试;模式中毒;路由劫持;WSDL 扫描
Key words: Web services; security testing; schema poisoning; routing detours; WSDL scanning
中图分类号:TP311　　**文献标识码**:A

1　引言

Web 服务作为一种分布式计算模型,以服务的形式封装应用并对外发布。Web 服务基于一系列开放的标准技术,具有松散耦合、语言中立、平台无关、互操作性的优点。越来越多的分布式应用采用 Web 服务作为其实现的技术基础。由于 Web 服务往往包含了企业关键业务,若其安全性出现问题,可能会造成重大损失与严重后果。Web 服务的安全性问题成为制约其广泛应用的主要障碍。Web 服务安全性测试是保证 Web 服务安全性的重要手段,但目前 Web 服务安全性相关研究大多局限在 Web 服务安全实现机制上,而从测试的角度进行研究还比较少。软件安全性测试是确定软件的安全特性实现是否与预期设计一致的过程,包括安全功能测试和安全漏洞测试。本文首先介绍 Web 服务安全性测试框架,然后从安全功能测试与漏洞测试两个方面对 Web 服务安全性测试技术进行一些探讨。

2　Web 服务安全性测试框架

Web 服务安全性测试框架用来指导 Web 服务安全性测试过程,可以减少测试活动的盲目性,提高测试活动的规范性,增强测试效能。如图 1 所示。

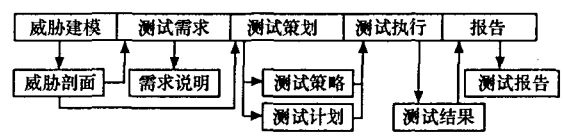

图 1　Web 服务安全性测试框架

Web 服务安全性测试框架包括五个阶段,分别是威胁建模、测试需求定义、测试策划、测试执行和报告。各阶段产生的文档主要有威胁剖面、测试需求说明书、测试策略、测试计划、测试结果与测试报告。威胁建模的主要步骤是

* 收稿日期:2007-03-29;修订日期:2007-07-09
作者简介:施寅生(1983-),男,河南商丘人,硕士生,研究方向为软件测试技术和 Web 服务安全技术;邓世伟,研究员,研究方向为软件测试和软件质量保证;谷天阳,硕士,研究方向为数据共享和软件测试技术。
通讯地址:100101 北京市 9702 信箱 19 号 3 室;Tel:(010)66356615;E-mail:shysh_1219@163.com
Address: Section 3, P. O. Box 9702-19, Beijing 100101, P. R. China

确定安全目标、创建应用程序总体体系结构、分解应用程序、确定威胁、确定漏洞、评定威胁与漏洞等级、威胁文档化。测试需求定义是根据软件的安全需求说明与威胁剖面文档来确定安全测试的对象、内容及测试资源的分配。测试策划主要确定测试策略与测试计划。测试策略文档描述应用程序总体架构、测试目标、拟采用的测试方法、资源需求(WSDL文档、源代码等)、缺陷跟踪与变更控制策略。测试计划描述测试环境要求、针对威胁与安全需求设计的测试用例、人员职责分配、进度安排、测试完成标准等。最后两个阶段记录测试结果并创建测试报告。

3 Web服务安全功能测试技术

表1列出了Web服务常见安全功能的实现标准。

表1 Web服务常见安全功能实现标准

安全功能需求	实现标准
身份认证	SAML, Kerberos, DSS, Generic Security Service(GSS), Liberty, WS-Security, WS-Federation
机密性	WS-Security, XML Encryption, XKMS, SSL/TLS, HTTPS
完整性	WS-Security, XML Signature
不可否认性	WS-Security, XML Signature
访问控制	XACML, JAAS, XrML
审计	NIST SP 800-92
隐私保护	WS-Privacy, P3P

安全功能测试基于软件的安全功能需求,测试软件安全功能实现是否与需求一致,需求实现是否正确完备[1]。Web服务主要的安全功能需求包括消息机密性、完整性、可用性、不可否认性、身份认证、授权、访问控制、审计跟踪、隐私保护、安全管理等。Web服务安全功能测试主要针对这些安全特性的具体实现进行,可能需要从传输级、消息级、应用级展开。Web服务传输级安全主要基于SSL、TLS、HTTPS、IPSec、VPN等实现。传输级安全机制可满足点到点的安全通信需求,但效率和性能不高,并难于实现跨中介体的端到端安全服务[2]。越来越多的Web服务采用基于WS-Security的消息级安全措施,实现端到端的安全通信。进行Web服务消息级安全功能测试需要生成大量的加密SOAP消息、签名SOAP消息、SAML断言消息、包含安全令牌(用户名令牌、X.509令牌、Kerberos票据令牌)的SOAP消息。针对Web服务消息机密性的测试是通过发送加密的SOAP消息观察其响应,判断机密性功能实现是否正确。例如,若Web服务要求使用X.509安全令牌加密SOAP消息,向该服务发送采用正确令牌加密的SOAP请求,返回正确的操作调用结果,说明其机密性功能正确实现。反之,若返回错误的调用结果或SOAP故障消息,说明其机密性功能没有正确实现。针对Web服务消息完整性的测试,通过发送签名的SOAP消息观察被测服务响应,判断完整性功能实现是否正确。对于实现WS-Security用户名令牌认证机制的Web服务,要测试其认证功能需要发送包含用户名安全令牌的SOAP消息。Web服务应用级安全由应用程序负责提供自定义的安全功能。例如,应用程序可以使用自定义的SOAP头传递用户凭证,以便根据每个Web服务请求对用户进行身份认证。进行Web服务安全功能测试需要了解Web服务安全功能实现技术、相关标准。

4 Web服务安全漏洞测试技术

Web服务采用了一系列开放的标准和技术,在最大程度实现跨平台、互操作、易访问的同时更易于受到攻击。这是由于Web服务SOAP消息常通过HTTP80端口传输,而80端口为大多数防火墙允许。网络级防火墙只过滤网络层数据包,应用级防火墙目前只过滤HTML流量。入侵检测系统只能捕获特定网络攻击。遗留应用包装成Web服务,存在很多固有脆弱性。Web服务消息是人可读的、自描述的,更易于泄露信息。Gartner组织的一份报告认为,Web服务重新开放了过去十年已被防火墙关闭的70%的攻击路径[3]。Web服务安全漏洞测试是从攻击者的角度,以发现Web服务的安全漏洞为目的。Web服务安全漏洞测试可采用模拟攻击测试的方法,利用已知漏洞攻击Web服务,测试其是否具有相应漏洞。

4.1 Web服务接口探查

4.1.1 WSDL扫描[4]

Web服务描述文件(WSDL文件)描述了服务提供的方法、调用方法需要的参数个数与参数类型等重要信息。WSDL扫描又称WSDL枚举,主要是扫描WSDL文件,列出可调用的方法,并根据已知方法推测未列出的方法。

4.1.2 参数篡改

参数篡改是利用WSDL扫描的结果,根据服务调用可接受的参数类型,故意发送非期望的数据类型尝试攻击Web服务。

4.2 攻击XML解析器

4.2.1 递归负载

XML消息可以进行递归实体扩展,恶意构造包含大量递归嵌套元素的消息,以耗尽服务器资源或使解析器崩溃,造成拒绝服务攻击。例如,可以构造递归嵌套100 000层的XML元素。递归负载还可以通过恶意DTD文档来定义。例如:

```
<? xml version="1.0" encoding="utf-8"? >
  <! DOCTYPE foobar [
    <! ENTITY x0 "hello">
    <! ENTITY x1 "&x0;&x0;">
    <! ENTITY x2 "&x1;&x1;">
    ...
    <! ENTITY x98 "&x97;&x97;">
    <! ENTITY x99 "&x98;&x98;">
    <! ENTITY x100 "&x99;&x99;">
  ]>
  <foobar>&x100;</foobar>
```

引用实体X100将产生2 100个hello字符串。基于DOM实现的解析器易受到递归负载攻击。执行漏洞测试可以人工或利用工具产生递归XML消息,攻击Web服务。

4.2.2 特大负载

XML解析器对于非常大的XML消息常常会处理出错。这种测试就是有意构造特大的SOAP消息,试图使解

析器耗尽内存及CPU资源，造成拒绝服务。例如，可以构造几百兆或更大的消息。

4.2.3 强制解析

这种测试通常是构造畸形的XML消息，试图使目标服务降级或不可用。例如，可以构造包含非常长的元素名称、非常多的标记或标记不匹配的消息。

4.3 基于内容的攻击

4.3.1 恶意SOAP附件

Web服务常常对SOAP附件缺乏验证，造成恶意代码嵌入附件中传播。例如，病毒、蠕虫等通过SOAP附件传播[5]。这种测试即是构造带恶意附件的SOAP消息发送到Web服务。

4.3.2 注入式攻击

注入式攻击包括SQL注入、XPATH注入、LDAP注入、XML注入、命令注入等。SOAP消息携带了服务调用需要的参数，这些参数可能是执行SQL查询或XPATH查询语句的一部分。恶意构造的用户输入有可能绕过数据库认证而执行未授权的查询或者恶意篡改数据库、执行系统命令等。以下消息中斜体部分为恶意构造的用户输入：

```
<SOAP:Envelope xmlns:SOAP=…>
 <SOAP:Header>…</SOAP:Header>
  <SOAP:Body
   <BookLookup:searchByIBSN xmlns:BookLookup=…>
   <BookLookup:IBSN>'exec master..xp_cmdshell' net
user Joe pass /ADD'</BookLookup:IBSN>
   </BookLookup:searchByIBSN>
  </SOAP:Body>
</SOAP:Envelope>
```

常见的代码是：

```
myRecordset= myConnection.execute("SELECT * FROM
myBooksTable WHERE IBSN = '" & IBSN_Element_Text
&"'")
```

转换为SQL语句是：

```
SELECT * FROM myTable WHERE IBSN = 'exec mas-
ter..xp_cmdshell'net user Joe pass /ADD
```

可以看出，一条命令成功注入。

XPATH注入类似。它是用来查询XML文档的语言。例如，以下XPATH语句用来查询用户名和密码：

```
//user[name='Joe'and pass='letmein']
```

通过注入'or 1=1 or''='，结果返回所有用户：

```
//user[name='Joe'or1=1or''=''and pass='letmein']
```

XPATH注入可能使攻击者访问未授权的文档信息。

XML注入就是通过注入XML元素或属性到XML消息中，改变原来消息的结构和语义。例如，以下斜体语句即为注入的XML语句：

```
<UserRecord>
<UniqueID>12345</UniqueID>
<Name>Henry Ackerman</Name>
<Email>hackerman@bad.com</Email><UniqueID>0
</UniqueID></Email>hackerman@bad.com</Email>
<Address>123 Disk Drive</Address>
<ZipCode>98103</ZipCode>
<PhoneNumber>206-123-4567</PhoneNumber>
</UserRecord>
```

4.3.3 跨站脚本

跨站脚本在Web应用中是常出现的漏洞，在Web服务中同样可能存在。Web服务中实现跨站脚本常利用CDATA节，因为CDATA节中的内容是不被解析的。以下是一个跨站脚本的例子：

```
<TAG1>
 <![CDATA[<]]>SCRIPT<![CDATA[>]]>
 alert('XSS');
 <![CDATA[<]]>/SCRIPT<![CDATA[>]]>
</TAG1>
<TAG2>
 <![CDATA['or 1=1 or''='']]>
</TAG2>
```

4.4 外部引用攻击

4.4.1 外部实体攻击

XML消息可以引用外部实体，但若Web服务对外部实体引用缺乏验证，可利用恶意外部实体嵌入恶意数据或系统命令攻击Web服务。

4.4.2 模式中毒

模式中毒[6]（Schema Poisoning）是通过操纵或修改XML模式文件，改变Web服务所接受消息的格式和语义，使Web服务接受不期望的数据类型等。

4.4.3 路由劫持

路由劫持[6]（Routing Detours）攻击是在SOAP头部修改或添加恶意路由目标，重定向SOAP消息到恶意接受者，然后接受者去除额外路由指令并转发SOAP消息，合法接收者无法察觉消息被篡改，造成敏感信息泄露。例如，以下消息中malicious.com是被添加的路由信息：

```
<wsrp:to>SOAP://endpoint.com<wsrp:to>
<wsrp:fwd>
<wsrp:via>SOAP://intermediary1.com</wsrp:via>
<wsrp:via>SOAP://MALICIOUS.com</wsrp.via>
<wsrp:via>SOAP://intermediary2.com</wsrp:via>
</wsrp:fwd>
<wsrp:from>SOAP://origin.com</wsrp:to>
```

4.5 不适当的错误处理

Web服务常常含有一类不太引人注意的脆弱性，就是返回的错误消息往往提供过于详细的信息，而这有助于攻击者发现应用程序结构等敏感信息，例如后端数据库的类型、采用的解析器等。因此，执行这类测试就是强制Web服务返回错误消息，观察错误消息的内容，分析是否存在不恰当的错误处理。

5 Web服务安全性测试工具

Web服务安全性测试技术研究尚处于起步阶段，相关支持工具还比较少，尚不成熟。商用工具主要有Parasoft公司的SOATest、Vordel公司的SOAPBox、CrossCheck公司的SOAPSonar、Forum Systems公司的XRay diagnosis、Optimyz公司的WebServiceTester、SPI Dynamics公司的WebInspect、Watchfire公司的AppScan。开源工具有WSDigger。这些工具在测试工程中基本充当两种角色：客户端（向目标服务发出请求）和代理服务（充当服务代理接收客户服务请求）。这些工具生成各种安全性测试数据对Web服务的安全功能进行测试，主要实现WS-Security安全规范的验证或者提供常见Web服务安全漏洞的测试支持。

（下转第28页）

平方向重叠距离相关。拼合后的效果图如图4所示。

图4 两张圆鱼眼图像拼合效果图

3 漫游空间编辑和浏览器设计

漫游空间的编辑是一个全景漫游系统实现的重要组成部分。通过漫游空间的编辑，能把本来互不相干的全景图像组织成用户可任意漫游的虚拟实景空间。主要依据漫游空间模型，漫游空间的编辑主要是生成各种链、赋予链属性等。其次，需要创造一种合适的空间文件结构把全景图像与链数据存入文件中，以备浏览器使用。浏览器是提供给最终用户漫游虚拟实景空间时所使用的。

4 浙江师范大学全景漫游系统

我们采用本文讨论的球面全景生成方法，开发了浙江师范大学的校园全景漫游系统。对该大学的校园环境、实验环境、硬件设施等场景拍摄两张圆鱼眼图像，合成单个场景的球面全景，即完成全景图的生成，从而实现单个场景的交互式漫游。单个场景可以实现自动漫游和手动控制漫游。浏览场景时，能够快速、流畅地控制显示任意观察方向和变焦的场景。基于JAVA APPLET技术，实现场景之间的链接和快速切换，在显示页面的旁边加入导航图，使得可以随时随地地链接到自己感兴趣的场景，而且不会迷失方向。浙江师范大学校园全景漫游系统主界面如图5所示。

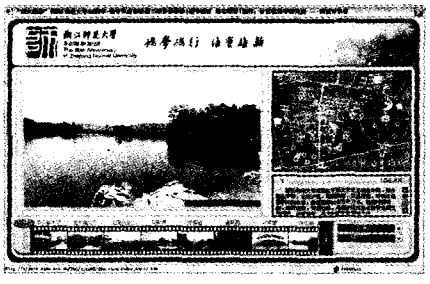

图5 校园漫游系统主界面

5 结束语

本文对球面全景图生成技术进行了研究，提出一种基于拼合参数自动寻优的球面全景图生成方法，并加以实现。该方法通过建立两个球缺模型，把待拼合的两张圆鱼眼图像映射到上下两个球缺模型上，通过求取两个球缺模型的三个拼合参数和两个球缺模型的并集来实现球面全景图的生成。针对两幅图像接痕明显问题，采用一种加权渐变的方法，使得两幅图像重叠部分自然平滑过渡。实验证明，该球面全景图生成方法具有所需源图像少、拼合速度快、拼合位置准确、生成全景图像质量高等优点。基于该方法，我们开发了浙江师范大学虚拟校园全景漫游系统。

参考文献：

[1] 汪嘉业，杨兴强，张彩明. 基于鱼眼镜头拍摄的图像生成漫游模型[J]. 系统仿真学报，2001，11(13)：66-68.

[2] 唐俊. 鱼眼图像轮廓提取算法研究[J]. 微机发展，2004，14(10)：9-14.

[3] 陈伟明，徐丹. 球面坐标定位校正鱼眼图片并合成全景图的方法[J]. 云南民族大学学报(自然科学版)，2004，13(3)：214-217.

[4] 英向华，胡占义. 一种基于球面透视投影约束的鱼眼镜头校正方法[J]. 计算机学报，2003，26(12)：1702-1708.

[5] 赵庆展，赵欣，常静. 虚拟校园全景漫游系统的实现[J]. 石河子大学学报(自然科学版)，2006，24(1)：124-126.

[6] 国家知识产权局专利文库[P]. 发明专利号：03115149.3，200410093187.0，200410093545.8.

(上接第13页)

6 结束语

随着Web服务应用的日益广泛和安全性问题的突出，Web服务的安全性测试在保证Web服务软件安全性方面的作用逐渐重要。但是，目前国内外相关研究还比较少，工具支持仍很薄弱。随着Web服务安全性相关标准规范的不断推出，Web服务安全实现技术日益成熟，机密性、完整性、认证、授权、隐私保护、审计等安全功能的测试方法与自动化支持工具尚亟待研究。随着Web服务相关漏洞的不断发布，Web服务安全漏洞测试方法与支持工具也必将成为未来的研究重点。

参考文献：

[1] Chandramouli R, Blackburn M R. Automated Testing of Security Functions Using a Combined Model and Interface-Driven Approach[A]. Proc of HICSS[C]. 2004.

[2] Kearney P. Message Level Security for Web Services[J]. Information Security Technical Report, 2005, (10): 41-50.

[3] Desmet L, Jacobs B, Piessens F, et al. Threat Modelling for Web Services Based Web Applications[A]. Proc of the 8th IFIP TC-6 TC-11 Conf on Communications and Multimedia Security[C]. 2004.

[4] Yu W D, Aravind D, Supthaweesuk P. Software Vulnerability Analysis for Web Services Software Systems[A]. Proc of the 11th IEEE Int'l Symp on Computers and Communications[C]. 2006. 740-748.

[5] Yunus M, Mallal R. An Empirical Study of Security Threats and Countermeasures in Web Services-Based Services Oriented Architectures[A]. Proc of the 6th Int'l Conf on Web Information Systems Engineering[C]. 2005. 653-659.

[6] Lindstrom P. Attacking and Defending Web Services[EB/OL]. http://forumsystems.com/papers/Attacking_and_Defending_WS.pdf, 2004-01.