安全性

八哥车间BBS论坛采用B/S架构，客户端软件主要是Web页面。

**安全性可分为: 数据或功能访问控制安全和页面访问控制安全。**

**数据或功能访问控制安全**

数据或功能访问安全性可保证在预期的安全性情况下，不同授权的用户只能访问特定的功能，或者只能访问有限的数据。例如，普通用户可进行输入数据、创建新账户等操作而不能随意删除数据或账户，只有管理员才能删除数据或账户。

**页面访问控制安全**

(1)页面登录。B/S 架构软件必须测试登录用户名和密码的有效性、输入大小写的敏感性、用户登录是否有次数限制; 而且要测试是否可以不登陆而直接浏览某个页面、IP地址登录是否有限制等;

(2) 超时限制。如果用户登陆系统后在一定时间 内没有进行任何页面操作，应进行超时判断，强制用户 重新登陆后才能正常使用系统。

(3) 日志文件。日志文件是保证 Web 应用系统安全性的重要的工具。B/S 架构软件需要测试日志文件 记录信息的完整性、各类操作的可追踪性。在服务器后台，还要检测服务器的日志记录是否正常进行。

(4) SSL。为保证信息在Internet 上传输的安全性，B/S架构软件会采用SSL技术。如果使用了SSL，测试人员需要测试加密的正确性、检测信息的完整性，确定是否有相应的替代页面。当用户进入或离开安全 站点的时候，是否有相应的提示信息。

(5)脚本安全性。服务器端的各类数据处理脚本常常成为安全漏洞而被黑客所利用。找出当前站点使用了哪些脚本语言，并研究该语言的缺陷，而且应该确定没有经过授权，相关人员就不能在服务器端保存和编辑相关脚本。

**安全性测试技术手段**

为确保数据访问安全而进行的**安全功能测试**和为保证页面访问安全而进行的**安全漏洞测试**。安全功能测试基于软件的安全功能需求说明，测试软件的安全功能实现是否与安全需求一致; 安全漏洞测试则站在攻击者的角度，以发现软件的安全漏洞为目的。

**安全功能测试**

Web服务主要的安全功能需求包括身份认证、消息机密性、不可否认性、完整性、授权、可用性、访问控制、审计跟踪、安全管理、隐私保护等。安全功能测试主要针对上述安全内容的具体实现进行功能性测试验证。功能验证就是对软件需求中确定的有关安全模块的功能进行测试验证。开发者一般会在软件设计和开发过程中增加一些必要的安全防护措施，如权限管理模块、数据加密模块、传输加密模块等。安全功能验证可以采用与一般程序功能测试相似的方法，如黑盒测试、白盒测试或灰盒测试等方法来进行。

**安全漏洞测试**

安全漏洞测试就是识别软件的安全漏洞。漏洞指的是存在于一个系统内的弱点或缺陷，漏洞被利用可能造成软件受到攻击，使软件进入不安全的状态。

B/S应用程序攻击包括: 对程序本身的拒绝服务攻击、改变网页内容以及窃取用户信息等。常见的B/S应用安全漏洞有: 已知弱点和错误配置、隐藏字段、后门和调试漏洞、跨站点脚本编写、参数篡改、更改 Cookies、缓冲区溢出、直接访问浏览等。

参数篡改

WSDL文件( Web服务描述文件) 保存了Web服务提供的方法、调用这些方法需要的参数个数以及需要的参数类型等重要信息。参数篡改是指: 首先对WSDL文件进行扫描，寻找Web 服务调用可接受的参数类型，故意发送Web服务不期望的数据类型，对Web服务进行攻击。

XML解析器攻击

XML消息可以对递归实体进行扩展。基于这一特征，攻击者通过恶意构造包含大量递归嵌套元素的消息，例如构造嵌套100000层的消息，用来耗尽服务器资源或者使XML解析器崩溃，达到拒绝服务攻击的目的。

注入式攻击

SOAP(简单对象访问协议) 是一种轻量的、简单的、基于XML的协议，它被设计成在 Web 上交换结构化的和固化的信息。SOAP消息携带了Web服务调用需要的参数，而这些参数极有可能是SQL查询语句或者XPATH查询语句的一部分。注入式攻击者构造相关的查询语句或者认证语句用来绕过数据库认证，从而执行非法查询操作、恶意篡改数据或者非法执行系统命令等。例如下面的XPATH语句用来在数据库中查询操作者的用户名和密码: user［name = ‘nare’and pass = ‘misback’］攻击者通过插入字段 or 1 = 1 or‘’ ，构造新语句，导致系统返回所有的用户信息: user［name = ‘nare’ or 1 = 1or‘’and pass = ‘mis-back’］

跨站脚本攻击

跨站脚本攻击(也称为XSS) 指利用网站漏洞从用户那里恶意盗取信息。用户在浏览网站、使用即时通讯软件或者在阅读电子邮件时，通常会点击其中的链接。攻击者通过在链接中插入恶意代码，就能够盗取用户信息。对于跨站脚本攻击进行防范主要有两方面: 验证所有输入数据，有效检测攻击; 对所有输出数据进行适当的编码，以防止任何已成功注入的脚本在浏览器端运行。