**Web网站测试方法**

在Web工程过程中，基于Web系统的测试、确认和验收是一项重要而富有挑战性的工作。基于Web的系统测试与传统的软件测试不同，它不但需要检查和验证是否按照设计的要求运行，而且还要测试系统在不同用户的浏览器端的显示是否合适。重要的是，还要从最终用户的角度进行安全性和可用性测试。然而，Internet和Web媒体的不可预见性使测试基于Web的系统变得困难。因此，我们必须为测试和评估复杂的基于Web的系统研究新的方法和技术。

　　本文将 web 测试分为 6 个部分：

　　1. 功能测试

　　2. 性能测试(包括负载/压力测试)

　　3. 用户界面测试

　　4. 兼容性测试

　　5. 安全测试

　　6. 接口测试

　　本文的目的是覆盖 web 测试的各个方面，未就某一主题进行深入说明。

　　1 功能测试

　　1.1 链接测试

　　链接是Web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。链接测试可分为三个方面。首先，测试所有链接是否按指示的那样确实链接到了该链接的页面;其次，测试所链接的页面是否存在;最后，保证Web应用系统上没有孤立的页面，所谓孤立页面是指没有链接指向该页面，只有知道正确的URL地址才能访问。

　　链接测试可以自动进行，现在已经有许多工具可以采用。链接测试必须在集成测试阶段完成，也就是说，在整个Web应用系统的所有页面开发完成之后进行链接测试。

　　采取措施：采用自动检测网站链接的软件来进行。

　　推荐软件：

　　Xenu Link Sleuth 免费 绿色免安装软件

　　HTML Link Validator 共享(30天试用)

　　1.2 表单测试

　　当用户通过表单提交信息的时候，都希望表单能正常工作。

　　如果使用表单来进行在线注册，要确保提交按钮能正常工作，当注册完成后应返回注册成功的消息。如果使用表单收集配送信息，应确保程序能够正确处理这些数据，最后能让顾客能让客户收到包裹。要测试这些程序，需要验证服务器能正确保存这些数据，而且后台运行的程序能正确解释和使用这些信息。

　　当用户使用表单进行用户注册、登陆、信息提交等操作时，我们必须测试提交操作的完整性，以校验提交给服务器的信息的正确性。例如：用户填写的出生日期与职业是否恰当，填写的所属省份与所在城市是否匹配等。如果使用了默认值，还要检验默认值的正确性。如果表单只能接受指定的某些值，则也要进行测试。例如：只能接受某些字符，测试时可以跳过这些字符，看系统是否会报错。

　　1.3 数据校验

　　如果系根据业务规则需要对用户输入进行校验，需要保证这些校验功能正常工作。例如，省份的字段可以用一个有效列表进行校验。在这种情况下，需要验证列表完整而且程序正确调用了该列表(例如在列表中添加一个测试值，确定系统能够接受这个测试值)。

　　在测试表单时，该项测试和表单测试可能会有一些重复。

　　1.2和1.3的采取措施：第一个完整的版本采用手动检查，同时形成WinRunner(QTP)脚本;回归测试以及升级版本主要靠WinRunner(QTP)自动回放测试。

　　1.4 cookies测试

　　Cookies通常用来存储用户信息和用户在某应用系统的操作，当一个用户使用Cookies访问了某一个应用系统时，Web服务器将发送关于用户的信息，把该信息以Cookies的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登陆等信息。

　　如果Web应用系统使用了Cookies，就必须检查Cookies是否能正常工作。测试的内容可包括Cookies是否起作用，是否按预定的时间进行保存，刷新对Cookies有什么影响等。如果在 cookies 中保存了注册信息，请确认该 cookie能够正常工作而且已对这些信息已经加密。如果使用 cookie 来统计次数，需要验证次数累计正确。

　　采取措施：

　　1 采用黑盒测试：采用上面提到的方法进行测试

　　2 采用查看cookies的软件进行(初步的想法)

　　可以选择采用的软件

　　IECookiesView v1.50

　　Cookies Manager v1.1

　　1.5 数据库测试

　　在Web应用技术中，数据库起着重要的作用，数据库为Web应用系统的管理、运行、查询和实现用户对数据存储的请求等提供空间。在Web应用中，最常用的数据库类型是关系型数据库，可以使用SQL对信息进行处理。

　　在使用了数据库的Web应用系统中，一般情况下，可能发生两种错误，分别是数据一致性错误和输出错误。数据一致性错误主要是由于用户提交的表单信息不正确而造成的，而输出错误主要是由于网络速度或程序设计问题等引起的，针对这两种情况，可分别进行测试。

　　采取措施：暂时没有更好的测试方法

　　考虑结合到1.2和1.3的测试中

　　1.6 应用程序特定的功能需求

　　最重要的是，测试人员需要对应用程序特定的功能需求进行验证。尝试用户可能进行的所有操作：下订单、更改订单、取消订单、核对订单状态、在货物发送之前更改送货信息、在线支付等等。这是用户之所以使用网站的原因，一定要确认网站能像广告宣传的那样神奇。

　　采取措施：深刻理解需求说明文档

　　1.7 设计语言测试

Web设计语言版本的差异可以引起客户端或服务器端严重的问题，例如使用哪种版本的HTML等。当在分布式环境中开发时，开发人员都不在一起，这个问题就显得尤为重要。除了HTML的版本问题外，不同的脚本语言，例如Java、JavaScript、 ActiveX、VBScript或Perl等也要进行验证

**4 兼容性测试**

**4.1 平台测试**

市场上有很多不同的操作系统类型，最常见的有Windows、Unix、Macintosh、Linux等。Web应用系统的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。这样，就可能会发生兼容性问题，同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。

　　因此，在Web系统发布之前，需要在各种操作系统下对Web系统进行兼容性测试。

**4.2 浏览器测试**

浏览器是Web客户端最核心的构件，来自不同厂商的浏览器对Java，、JavaScript、 ActiveX、 plug-ins或不同的HTML规格有不同的支持。例如，ActiveX是Microsoft的产品，是为Internet Explorer而设计的，JavaScript是Netscape的产品，Java是Sun的产品等等。另外，框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不显示。不同的浏览器对安全性和Java的设置也不一样。

　　测试浏览器兼容性的一个方法是创建一个兼容性矩阵。在这个矩阵中，测试不同厂商、不同版本的浏览器对某些构件和设置的适应性。

**4.3 分辨率测试**

页面版式在640x400、600x800或1024x768 的分辨率模式下是否显示正常? 字体是否太小以至于无法浏览? 或者是太大? 文本和图片是否对齐?

**4.4 Modem/连接速率**

是否有这种情况，用户使用28.8 modem下载一个页面需要10分钟，但测试人员在测试的时候使用的是T1专线? 用户在下载文章或演示的时候，可能会等待比较长的时间，但却不会耐心等待首页的出现。最后，需要确认图片不会太大。

**4.5 打印机**

用户可能会将网页打印下来。因此网也在设计的时候要考虑到打印问题，注意节约纸张和油墨。有不少用户喜欢阅读而不是盯着屏幕，因此需要验证网页打印是否正常。有时在屏幕上显示的图片和文本的对齐方式可能与打印出来的东西不一样。测试人员至少需要验证订单确认页面打印是正常的。

**4.6 组合测试**

最后需要进行组合测试。600x800的分辨率在 MAC 机上可能不错，但是在IBM 兼容机上却很难看。在IBM机器上使用Netscape能正常显示，但却无法使用Lynx 来浏览。如果是内部使用的web 站点，测试可能会轻松一些。如果公司指定使用某个类型的浏览器，那么只需在该浏览器上进行测试。如果所有的人都使用 T1 专线，可能不需要测试下载施加。(但需要注意的是，可能会有员工从家里拨号进入系统) 有些内部应用程序，开发部门可能在系统需求中声明不支持某些系统而只支持一些那些已设置的系统。但是，理想的情况是，系统能在所有机器上运行，这样就不会限制将来的发展和变动。

采取措施：根据实际情况，采取等价划分的方法，列出兼容性矩阵

**5 安全测试**

即使站点不接受信用卡支付，安全问题也是非常重要的。Web 站点收集的用户资料只能在公司内部使用。如果用户信息被黑客泄露，客户在进行交易时，就不会有安全感。

**5.1 目录设置**

Web 安全的第一步就是正确设置目录。每个目录下应该有 index.html 或 main.html 页面，这样就不会显示该目录下的所有内容。我服务的一个公司没有执行这条规则。我选中一幅图片，单击鼠标右键，找到该图片所在的路径"…com/objects/images"。然后在浏览器地址栏中手工输入该路径，发现该站点所有图片的列表。这可能没什么关系。我进入下一级目录 "…com/objects" ，点击 jackpot。在该目录下有很多资料，其中引起我注意的是已过期页面。该公司每个月都要更改产品价格，并且保存过期页面。我翻看了一下这些记录，就可以估计他们的边际利润以及他们为了争取一个合同还有多大的降价空间。如果某个客户在谈判之前查看了这些信息，他们在谈判桌上肯定处于上风。

**5.2 SSL**

很多站点使用 SSL 进行安全传送。你知道你进入一个 SSL 站点是因为浏览器出现了警告消息，而且在地址栏中的 HTTP 变成 HTTPS。如果开发部门使用了SSL，测试人员需要确定是否有相应的替代页面(适用于3.0 以下版本的浏览器，这些浏览器不支持SSL。当用户进入或离开安全站点的时候，请确认有相应的提示信息。是否有连接时间限制？超过限制时间后出现什么情况？

**5.3 登录**

有些站点需要用户进行登录，以验证他们的身份。这样对用户是方便的，他们不需要每次都输入个人资料。你需要验证系统阻止非法的用户名/口令登录，而能够通过有效登录。用户登录是否有次数限制? 是否限制从某些 IP 地址登录? 如果允许登录失败的次数为3，你在第三次登录的时候输入正确的用户名和口令，能通过验证吗? 口令选择有规则限制吗?  是否可以不登陆而直接浏览某个页面？

Web应用系统是否有超时的限制，也就是说，用户登陆后在一定时间内（例如15分钟）没有点击任何页面，是否需要重新登陆才能正常使用。

**5.4 日志文件**

在后台，要注意验证服务器日志工作正常。日志是否记所有的事务处理? 是否记录失败的注册企图? 是否记录被盗信用卡的使用? 是否在每次事务完成的时候都进行保存? 记录IP 地址吗? 记录用户名吗?

**5.5 脚本语言**

脚本语言是常见的安全隐患。每种语言的细节有所不同。有些脚本允许访问根目录。其他只允许访问邮件服务器，但是经验丰富的黑客可以将服务器用户名和口令发送给他们自己。找出站点使用了哪些脚本语言，并研究该语言的缺陷。还要需要测试没有经过授权，就不能在服务器端放置和编辑脚本的问题。最好的办法是订阅一个讨论站点使用的脚本语言安全性的新闻组。

**6 接口测试**

在很多情况下，web 站点不是孤立。Web 站点可能会与外部服务器通讯，请求数据、验证数据或提交订单。

**6.1服务器接口**

第一个需要测试的接口是浏览器与服务器的接口。测试人员提交事务，然后查看服务器记录，并验证在浏览器上看到的正好是服务器上发生的。测试人员还可以查询数据库，确认事务数据已正确保存。

这种测试可以归到功能测试中的表单测试和数据校验测试中

**6.2 外部接口**

有些 web 系统有外部接口。例如，网上商店可能要实时验证信用卡数据以减少欺诈行为的发生。测试的时候，要使用 web 接口发送一些事务数据，分别对有效信用卡、无效信用卡和被盗信用卡进行验证。如果商店只使用 Visa 卡和Mastercard卡，可以尝试使用 Discover 卡的数据。(简单的客户端脚本能够在提交事务之前对代码进行识别，例如 3 表示 American Express，4 表示 Visa，5 表示Mastercard，6 代表Discover。)通常，测试人员需要确认软件能够处理外部服务器返回的所有可能的消息。

这种情况在远程抄表中可能会体现到

**6.3 错误处理**

最容易被测试人员忽略的地方是接口错误处理。通常我们试图确认系统能够处理所有错误，但却无法预期系统所有可能的错误。尝试在处理过程中中断事务，看看会发生什么情况？订单是否完成？尝试中断用户到服务器的网络连接。尝试中断 web 服务器到信用卡验证服务器的连接。在这些情况下，系统能否正确处理这些错误？是否已对信用卡进行收费？如果用户自己中断事务处理，在订单已保存而用户没有返回网站确认的时候，需要由客户代表致电用户进行订单确认。

采取措施：在理解需求的基础上，充分发挥想象力，尽量比较全面的列出各种异常情况

**Web网站测试流程**

1测试流程与方法
1.1测试流程
进行正式测试之前，应先确定如何开展测试，不可盲目的测试。一般网站的测试，应按以下流程来进行：
1）使用HTML Link Validator将网站中的错误链接找出来；
2）测试的顺序为：自顶向下、从左到右；
3）查看页面title是否正确。（不只首页，所有页面都要查看）；
4）LOGO图片是否正确显示；
5）LOGO下的一级栏目、二级栏目的链接是否正确；
6）首页登录、注册的功能是否实现；
7）首页左侧栏目下的文章标题、图片等链接是否正确；
8）首页中间栏目下的文章标题、图片等链接是否正确；
9）首页右侧栏目下的文章标题、图片等链接是否正确；
10）首页最下方的【友情链接】、【关于我们】等链接是否正确；
11）进入一级栏目或二级栏目的列表页。查看左侧栏目名称，右侧文章列表是否正确；
12）列表页的分页功能是否实现、样式是否统一；
13）查看文章详细页面的内容是否存在乱码、页面样式是否统一；
14）站内搜索（各个页面都要查看）功能是否实现；
15）前后台交互的部分，数据传递是否正确；
16) 默认按钮要支持Enter及选操作，即按Enter后自动执行默认按钮对应操作。
1.2 UI测试
UI测试包括的内容有如下几方面：
1）各个页面的样式风格是否统一；
2）各个页面的大小是否一致；同样的LOGO图片在各个页面中显示是否大小一致；页面及图片是否居中显示；
3）各个页面的title是否正确；
4）栏目名称、文章内容等处的文字是否正确，有无错别字或乱码；同一级别的字体、大小、颜色是否统一；
5）提示、警告或错误说明应清楚易懂，用词准确，摒弃模棱两可的字眼；
6）切换窗口大小，将窗口缩小后，页面是否按比例缩小或出现滚动条；各个页面缩小的风格是否一致，文字是否窜行；
7）父窗体或主窗体的中心位置应该在对角线焦点附近；子窗体位置应该在主窗体的左上角或正中；多个子窗体弹出时应该依次向右下方偏移，以显示出窗体标题为宜；
8）按钮大小基本相近，忌用太长的名称，免得占用过多的界面位置；避免空旷的界面上放置很大的按钮；按钮的样式风格要统一；按钮之间的间距要一致；
9）页面颜色是否统一；前景与背景色搭配合理协调，反差不宜太大，最好少用深色或刺目的颜色；
10）若有滚动信息或图片，将鼠标放置其上，查看滚动信息或图片是否停止；
11）导航处是否按相应的栏目级别显示；导航文字是否在同一行显示；
12）所有的图片是否都被正确装载，在不同的浏览器、分辨率下图片是否能正确显示（包括位置、大小）；
13）文章列表页，左侧的栏目是否与一级、二级栏目的名称、顺序一致；
14) 调整分辨率验证页面格式是否错位现象；
15）鼠标移动到Flash焦点上特效是否实现，移出焦点特效是否消失；
16) 文字颜色与页面配色协调，不使用与背景色相近的颜色。
17) 每个非首页静态页面含图片字节不超过300K，全尺寸banner第一个场景控制在200k以内 二个场景在300K，三个场景在400K以此类推
18) 同一界面上的控件数最好不要超过10个，多于10个时可以考虑使用分页界面显示。
19) 超过一屏的内容，在底部应有go top按钮
20) 超过三屏的内容，应在头部设提纲，直接链接到文内锚点
21) 首页，各栏目一级页面之间互链，各栏目一级和本栏目二级页面之间互链
22) 导航的文字要简明扼要，字数限制在一行以内
23) 报表显示时应考虑数据显示宽度的自适应或自动换行。
24) 所有有数据展现的界面（如统计、查询、编辑录入、打印预览、打印等），必须使测试数据的记录数超过一屏/一页，以验证满屏/页时其窗体是否有横向、纵向滚动条或换页打(L)印，界面显示是否正常；
25) 如有多个系统展现同一数据源时，应保证其一致性；
26) 对于报表中的所有字段值都应该有明确的定义，对于无意义的字段值，不应该显示空，应显示“--”或“/”，表示该字段值无意义。
27) 对统计的数据应按用户习惯进行分类、排序。
28) 界面内容更新后系统应提供刷新功能。
29) 用户在退出系统后重新登陆时应考虑是否需要自动返回到上次退出系统时的界面；
30)在多个业务功能组成的一个业务流程中，如果各个功能之间的执行顺序有一定的制约条件，应通过界面提示用户。
31)用户提示信息应具有一定的指导性，在应用程序正在进行关键业务的处理时，应考虑在前台界面提示用户应用程序正在进行的处理，以及相应的处理过程，在处理结束后再提示用户处理完毕。
32)在某些数据输入界面，如果要求输入的数据符合某项规则，应在输入界面提供相应的规则描述；当输入数据不符合规则时应提示用户是否继续。
33)在对任何配置信息修改后，都应该在用户退出该界面时提示用户保存（如果用户没有主动保存的情况下）；
34)在对某些查询功能进行测试时，应考虑查询条件的设置的合理性以及查询结果的互补性。如某些后台处理时间不应该作为查询条件。
35)界面测试时，应考虑某一界面上按钮先后使用的顺序问题，以免用户对此产生迷惑。例如只能在查询成功后显示执行按钮。
36)界面测试时，应验证窗口与窗口之间、字段与字段之间的浏览顺序是否正确；
37)在某些对数据进行处理的操作界面，应考虑用户可能对数据进行处理的频繁程度和工作量，考虑是否可以进行批量操作。
38)界面测试时应验证所有窗体中的对象状态是否正常，是否符合相关的业务规则需要。
49)应验证各种对象访问方法（Tab 健、鼠标移动和快捷键）是否可正常使用，并且在一个激活界面中快捷键无重复；
40)界面测试不光要考虑合理的键盘输入，还应考虑是否可以通过鼠标拷贝粘贴输入。
41)对于统计查询功能的查询结果应验证其是否只能通过界面上的查询或刷新按键人工触发，应避免其他形式的触发。
42)对界面上的任何对象进行拖拉，然后进行查询、打印，应保证查询打印结果不变；
43)确保数据精度显示的统一：如单价0元，应显示为0.00元；
44)确保时间及日期显示格式的统一；
45)确保相同含义属性/字段名的统一；
46)对所有可能产生的提示信息界面内容和位置进行验证，确保所有的提示信息界面应居中。

1.3链接测试
链接测试主要分为以下几个方面：
1）页面是否有无法连接的内容；图片是否能正确显示，有无冗余图片，代码是否规范，页面是否存死链接（可以用HTML Link Validator工具查找）；
2）图片上是否有无用的链接；点击图片上的链接是否跳转到正确的页面；
3）首页点击LOGO下的一级栏目或二级栏目名称，是否可进入相应的栏目；
4）点击首页或列表页的文章标题的链接，是否可进入相应的文章的详细页面；
5）点击首页栏目名称后的【更多】链接，是否正确跳转到相应页面；
6）文章列表页，左侧的栏目的链接，是否可正确跳转到相应的栏目页面；
7）导航链接的页面是否正确；是否可按栏目级别跳转到相应的页面；
（例：【首页->服务与支持->客服中心】，分别点击“首页”、“服务与支持”、“客服中心”，查看是否可跳转到相应页面；）
8) 新闻、信息类内容通常用新开窗口方式打开。
9) 顶部导航、底部导航通常采取在本页打开。
1.4搜索测试
搜索测试主要分为以下几个方面：
1）搜索按钮功能是否实现；
2）输入网站中存在的信息，能否正确搜索出结果；
3）输入键盘中所有特殊字符，是否报错；特别关注：\_ ？ ’ . • \  / -- ；特殊字符
4）系统是否支持键盘回车键、Tab键；
5）搜索出的结果页面是否与其他页面风格一致；
6）在输入域输入空格，点击搜索系统是否报错；
7）本站内搜索输入域中不输入任何内容，是否搜索出的是全部信息或者给予提示信息；
8）精确查询还是模糊查询，如果是模糊查询输入：中%国。查询结果是不是都包含中国两个字的信息；
9）焦点放置搜索框中，搜索框内容是否被清空；
10）搜索输入域是否实现回车键监听事件；
1.5表单测试
表单测试主要分为以下几个方面：
1）注册、登录功能是否实现；
2）提交、清空按钮功能是否实现；
3）修改表单与注册页面数据项是否相同，修改表单是否对重名做验证；
4）提交的数据是否能正确保存到后台数据库中（后台数据库中的数据应与前台录入内容完全一致，数据不会丢失或被改变）；
5）表单提交，删除，修改后是否有提示信息；提示、警告、或错误说明应该清楚、明了、恰当。
6）浏览器的前进、后退、刷新按钮，是否会造成数据重现或页面报错；
7）提交表单是否支持回车键和Tab键；Tab键的顺序与控件排列顺序要一致，目前流行总体从上倒下，同时行间从左到右的方式
8）下拉列表功能是否实现和数据是否完整（例如：省份和市区下拉列表数据是否互动）；
1.6输入域测试
输入域测试主要分为以下几个方面：
1）对于手机、邮箱、证件号等的输入是否有长度及类型的控制；
2）输入中文、英文、数字、特殊字符（特别注意单引号和反斜杠）及这四类的混合输入，是否会报错；
3）输入空格、空格+数据、数据+空格，是否报错；
4）输入html语言的<head>，是否能正确显示；
5）输入全角、半角的英文、数字、特殊字符等，是否报错；
6）是否有必填项的控制；不输入必填项，是否有友好提示信息；
7）输入超长字段，页面是否被撑开；
8）分别输入大于、等于、小于数据表规定字段长度的数据，是否报错；
9）输入非数据表中规定的数据类型的字符，是否有友好提示信息；
10）在文本框中输入回车键，显示时是否回车换行；
11) 非法的输入或操作应有足够的提示说明。
1.7分页测试
分页测试主要分为以下几个方面：
1）当没有数据时，首页、上一页、下一页、尾页标签全部置灰；
2）在首页时，“首页”“上一页”标签置灰；在尾页时，“下一页”“尾页”标签置灰；在中间页时，四个标签均可点击，且跳转正确；
3）翻页后，列表中的数据是否扔按照指定的顺序进行了排序；
4）各个分页标签是否在同一水平线上；
5）各个页面的分页标签样式是否一致；
6）分页的总页数及当前页数显示是否正确；
7）是否能正确跳转到指定的页数；
8）在分页处输入非数字的字符（英文、特殊字符等），输入0或超出总页数的数字，是否有友好提示信息；
9）是否支持回车键的监听；
1.8 交互性数据测试
1）前台的数据操作是否对后台产生相应正确的影响
   （如：查看详细信息时，需扣除用户相应的授权点数）；
2）可实现前后台数据的交互（如：在线咨询，能否实现数据的交互实时更新）；数据传递是否正确；前后台大数据量信息传递数据是否丢失（如500个字符）；多用户交流时用户信息控制是否严谨；
3）用户的权限，是否随着授权而变化；
4）数据未审核时，前台应不显示；审核通过后，前台应可显示该条数据；

功能测试中还需注意以下几点内容：
1）点击【收藏我们】，标题是否出现乱码；收藏的url与网站的url是否一致；能否通过收藏夹来访问网站；
2）对于修改、删除等可能造成数据无法恢复的操作必须提供确认信息，给用户放弃选择的机会；
3）在文章详细页面，验证字体大小改变、打印、返回、关闭等功能是否实现；
2安全性测试
2.1目录设置
目录测试主要分为以下几个方面：
1）在测试路径上出现：<http://218.61.30.17:7001/dzgh/xwzx/khzl/2008/11/13/58127.html> 把/2008/11/13/58127.html去掉，看是否能出现目录下文件；
2）访问文件目录如果出现403错误，说明网页加以限制拒绝访问；
3)访问文件目录如果出现SSH其他根目录路径，说明有漏洞缺陷；
4）用X-Scan-v3.2-cn工具对网站服务器扫描。可以对网站参透出开启的端口号，SSH弱口令，网站是否存在高风险；比如:在扫描参数中输入测试网站的地址，点击扫描。如果扫描出网站端口号高风险或SSH弱口令可以与开发人员沟通进行修改；
5）测试有效和无效的用户名和密码，要注意到是否大小写敏感，可以试多少次的限制，是否可以不登陆而直接浏览某个页面等。
6）Web应用系统是否有超时的限制，也就是说，用户登陆后在一定时间内（例如15分钟）没有点击任何页面，是否需要重新登陆才能正常使用。
7）为了保证Web应用系统的安全性，日志文件是至关重要的。需要测试相关信息是否写进了日志文件、是否可追踪。
8）当使用了安全套接字时，还要测试加密是否正确，检查信息的完整性。
9）服务器端的脚本常常构成安全漏洞，这些漏洞又常常被黑客利用。所以，还要测试没有经过授权，就不能在服务器端放置和编辑脚本的问题。
10）网页加载速度测试可以采用HttpWatch软件等，可以知道那些内容影响网站整体速度。