**F6+++++++++++++++++++++++++1+`+`bnm,/·如何更有效使用 Rational AppScan 扫描大型网站，第 2 部分: 案例分析**

*Rational AppScan 工作原理和使用经验总结和讨论*

本文针对大型网站的特点，结合 Rational AppScan 的工作原理，分析从扫描策略选择，扫描设置，任务分解等角度来如果更有效使用 Rational AppScan 进行应用安全扫描。“知己知彼，百战不殆”，在第一部分中，已经讨论分析了 Rational AppScan 的工作原理和并分析大型网站的特点；在第二部分中将结合一个案例，看如何使用 Rational AppScan 来完成对大型网站的安全检查。

[查看本系列更多内容](http://www.ibm.com/developerworks/cn/views/rational/libraryview.jsp?search_by=%E5%A6%82%E4%BD%95%E6%9B%B4%E6%9C%89%E6%95%88%E4%BD%BF%E7%94%A8+Rational+AppScan+%E6%89%AB%E6%8F%8F%E5%A4%A7%E5%9E%8B%E7%BD%91%E7%AB%99) | 1http://dw1.s81c.com/developerworks/i/v17/dw-cmts-arrow.png [评论：](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "icomments)

[庄 俊乾](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "authorN1002D), 客户技术专家, IBM

[关闭 [x]](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "close)

庄俊乾，是 IBM 中国有限公司软件部 Rational 客户技术专家。专著方向为软件自动化测试，性能测试，应用安全测试等领域。为电信金融行业等多家公司提供软件过程管理的咨询、技术和产品实施服务工作。当前的主要研究方向有软件质量管理/应用安全测试等。您可以通过 developerWorks 社区与[庄俊乾](https://www.ibm.com/developerworks/mydeveloperworks/profiles/html/profileView.do?key=ebb8ff54-1cc7-42a9-8773-d7a4798ce8aa)进行交流。

2011 年 7 月 29 日

* [expand](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/#toggle)**内容**
  + [使用 AppScan 进行扫描](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "1.使用 AppScan 进行扫描)
  + [准备阶段](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N100AE)
  + [计划阶段](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N10112)
  + [执行阶段](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N102C7)
  + [检查阶段](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N102FB)
  + [分析阶段](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N1030A)
  + [案例分析](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "2.案例分析)
  + [总结](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "3.总结)
  + [参考资料](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "resources)
  + [评论](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "icomments)

| **下载试用版：[IBM® Rational® AppScan 标准版](http://www.ibm.com/developerworks/cn/downloads/r/appscan/)  |  [Web 应用安全与 IBM Rational AppScan 工具包](http://www.ibm.com/developerworks/cn/rational/kits/appscankit/)** |
| --- |
| 下载更多的 [IBM 软件试用版](http://www.ibm.com/developerworks/cn/downloads/)，并加入 [IBM 软件下载与技术交流群组](https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=38997896-bb16-451a-aa97-189a27a3cd5a/?lang=zh)，参与在线交流。 |

**使用 AppScan 进行扫描**

针对大型网站的扫描，我们按照戴明环 PDCA 的方法论来进行规划和讨论，建议 AppScan 使用步骤：计划（Plan）、执行（Do）、检查（check）、分析（Analysis and Action）。

1. 在计划阶段：明确目的，进行策略性的选择和任务分解。
   1. 明确目的：选择合适的扫描策略
   2. 了解对象：首先进行探索，了解网站结构和规模
   3. 确定策略：进行对应的配置
   4. 按照目录进行扫描任务的分解
   5. 按照扫描策略进行扫描任务的分解
2. 执行阶段：一边扫描一遍观察
   1. 进行扫描
   2. 先爬后扫（继续仅测试）
3. 检查阶段（Check）
   1. 检查和调整配置
4. 结果分析（Analysis）
   1. 对比结果
   2. 汇总结果（整合和过滤）

下面我们针对每个阶段，进行具体的阐述。

**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**准备阶段**

**AppScan 安装环境要求和检查**

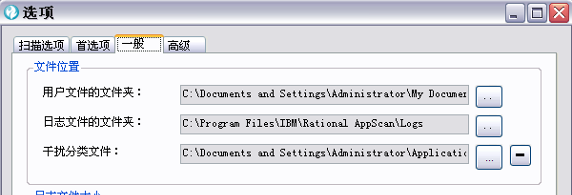
为了保证更好的扫描效果，安装 AppScan 的硬件建议配置如下：

**表 1. Rational AppScan 安装配置要求**

| **硬件** | **最低需求** |
| --- | --- |
| 处理器 | Pentium P4，2.4 GHz |
| 内存 | 2 GB RAM |
| 磁盘空间 | 30 GB |
| 网络 | 1 NIC 100 Mbps（具有已配置的 TCP/IP 的网络通信） |

其中，处理器和内存建议越大越好，而磁盘空间，建议系统盘（一般是 C 盘）磁盘空间至少保留 10G，如果系统盘磁盘空间比较少，可以考虑把用户文件等保存在其他盘；如默认的用户文件是：C:\Documents and Settings\Administrator\My Documents\AppScan；可以修改为其他路径。该路径可以在菜单栏中依次选择**工具 - 选项 - 一般 - 文件位置**部分修改。

**图 1. 设置文件保存路径**



**磁盘要求：修改临时文件路径**

有时候大家会发现，已经把上面的地址都修改到了其他盘，但是在扫描过程中，还是会发现 C 盘的空间快速被消耗，分析原因，是因为**很多临时文件都保存在 C 盘**，AppScan 中有一个隐藏的参数 APPSCAN\_TEMP 来设置临时文件位置。在扫描过程中，如果系统盘空间比较下，可以通过修改系统变量来修改到其他硬盘空间。

临时文件位置说明：描述正常操作期间 AppScan 将其临时文件保存到的位置。缺省情况下，AppScan 将其临时文件存储在以下位置：

C:\Documents and Settings\All Users\Application Data\IBM\Rational AppScan\temp

如果需要修改此缺省位置，请按照要求编辑环境变量 APPSCAN\_TEMP 的路径。（访问环境变量的方法是，右键单击我的电脑，然后依次选择**属性 > 高级 > 环境变量**。）

**注意：在新位置的路径中绝不能有任何 Unicode 字符。**

修改 AppScan 中的临时文件：

1. 桌面上鼠标右键选择“我的电脑”，选择“属性”
2. 选择“高级”，“环境变量”
3. 增加一个新的“用户环境变量”，名字是“APPSCAN\_TEMP”，设定路径，指向您希望保存临时文件的目录。

**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**计划阶段**

在计划阶段，首先明确几个问题：

1. 关心哪些类型的安全问题，根据这些安全问题来设置扫描规则。
2. 要扫描的网站地址，网站的业务特点。

**扫描策略的选择**

试想，我们现在要扫描的是某个移动公司的网站系统，该网站系统提供多个内容频道，还可以连接到多个其他移动公司网站和业务网站，我们本次安全测试重点关心的是门户网站本身和其上面的网上营业厅业务。这就是一个比较明确的测试目标对象。

然后，确定扫描策略，我们主要关心该网站是否存在跨站点脚本执行和 SQL 注入的问题，则在扫描规则中，我们就可以选择这两种类型的规则，其他规则都排除。

具体的扫描规则定制，可以在**扫描配置 - 测试 - 测试策略**中选择：

在测试策略中，有多种不同的分组模式，最经常使用的是“严重性”，“类型”，“侵入式”、“WASC 威胁分类”等标准，根据不同分组选择的扫描策略，最后组成一个共同的策略集合。

根据我们这次扫描的目标，关心的是跨站点脚本执行和 SQL 注入的问题，而且不考虑“基础结构”级别的安全问题。则就可以首先选择一个默认的扫描策略，然后全部置空，再选择

跨站点脚本执行和 SQL 注入，最后再去除这两种扫描策略中和基础结果相关的安全问题。

方法如下：

1. 选择缺省的扫描策略，或者把当前的扫描策略，切换到按照“**类型**”分类，取消掉“**基础结构**”和“**应用程序**”两种类型。

说明：则把扫描策略置空，没有选择任何的扫描策略，指所有分布类型选择“类型”分类，是因为类型分类里面含有的类型，只有两种类型，可以快速全部都取消掉。

1. 分组类型，切换到“**WASC 威胁分类**”，选择“**SQL 注入**”和“**跨站点脚本编制**”。
2. 分组类型，切换到“**类型**”，发现这时候“基础结构”和“应用程序”两种类型的扫描策略都是选择上的模式，而且是虚线，说明这两种类型下均有部分扫描策略被选择了。
3. 我们不关心“基础结构”级别的安全问题，所以在这里**取消“基础结构”**。

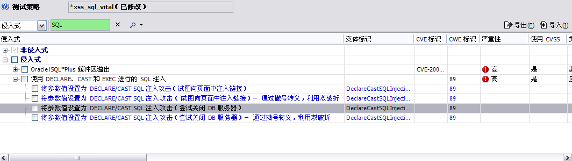
**图 2. 按“类型”分类的测试策略**



1. 分组类型，切换到“侵入式”类型，下面有“非侵入式”和“侵入式”两种分类。取消“基础结构”级别的测试。

**侵入式的测试用例**，往往因为有比较强的副作用，可能对系统造成伤害，所以一般扫描生产系统的时候，很少选择。我们可以查看一个 SQL 注入类型的侵入式安全问题，在“输入以查找”输入框中输入“SQL”，然后回车查询。可以看到测试变体的描述“将参数值设置为 Declare/Case SQL 注入攻击（尝试关闭 DB 服务器）”，则扫描过程中，会使用该测试用例去执行尝试关闭数据库的命令，如果该测试用例执行通过，则就关闭了数据库，则整个系统就瘫痪！所以，要很慎重的选择“侵入式的测试用例”。

**图 3. 查询测试策略**



[图 3 大图](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/origin_image003.png" \t "_blank)

其他的在“类型”中，“应用程序”类型表示该问题的存在是因为应用程序不严谨，代码存在安全问题而造成的，修改方法就是修改原代码；而“基础结构”类型，则表示该问题是配置问题，建议修改系统配置或者安装最新的补丁（经常是中间件或数据库补丁）。

**了解被测试网站**

在对网站进行测试之前，我们经常需要先大概了解下这个网站，比如该网站使用了哪些技术，提供什么类型的业务（功能），网站规模等。这些都和我们的扫描设置相关。如下图，就是我们经常使用的一个调查表，了解被测试系统的基本特点。

**表 2. 记录被测网站特点**

| **应用系统名称** | **访问地址** | **应用系统架构（JEE/.Net/PHP…）** | **URL 数量** | **登陆方式** | **备注** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

其中，用户经常迷惑的是 URL 数量，有些时候，用户很难评估出一个系统的大概页面数量，而按照 AppScan 的工作原理，扫描是针对页面的每个参数的，如果页面越多，参数越多，则扫描要运行的时间也就越长，扫描保存成的接过文件也是越大，更需要进行分解。**如果一个扫描任务，本身的已访问 URL 数超过 5000，评估的要运行的安全测试用例数超过 50，000，则建议进行扫描配置的分析，并根据分析结果，决定是否需要进一步的任务分解和分工。**

那么，如果可以了解到网站具体有哪些页面呢？这里我们就可以利用 AppScan 的探索（页面爬行）能力。

在扫描配置里面设置了主 URL 以后，工作菜单中中依次选择**扫描 - 仅探索**。对网站进行探索。一般会让探索工具运行 10 到 30 分钟，看该网站具体存在哪些页面，哪些参数等。这个就可以切换到“**应用程序数据**”视图来查看。

我们一般关心这几个视图：

* **已访问的 URL()**：AppScan 已经探索到并且进行了分析的页面
* **已过滤掉的 URL()**：AppScan 已经发现，同时根据扫描配置，认为不需要进行安全扫描的页面。
* **中断链接 URL()**：AppScan 发现了，但是无法访问到或者访问出错的页面，如 404 页面不存在，或者 500 服务器错误等。

**伪静态页面**

可以选择左边“**我的应用程序数据**”中的 URL 树下的每一个节点，察看该节点已访问的 URL，已过滤掉的 URL 等。

如在已访问的 URL() 中，我们发现大量类似如下结构的 HTML 页面：

http://www.Test.com//focus/satisfy/file5.html

http://www.Test.com//focus/satisfy/file6.html

http://www.Test.com/m-zone/news/dgdd/quanbu/bylb/file5.html

其共同特征，都是以 html 为后缀名，最后的文件名格式都是 **file+ 数字**格式；这种类型的页面经常存在新闻，论坛等。如果访问这些页面，发现页面结构相同，差异的都是里面的文本内容，如提供不同的新闻内容等，这些页面就是所谓的“伪静态页面”，其实是网站发布系统动态产生的，由于结果相似，在安全扫描中，没有必要针对这些页面每次都进行扫描。如针对每个目录下面存在的 file+ 数字格式的页面，我们就可以设置正则表达式来过滤，比如，在扫描配置 - 排除路径和文件中

*排除所有该类型的页面；.\*file\d+.html*

*增加“例外”，对该类型的页面只扫描 file1.html 和 file20.html*

经常存在的其他类似页面，还有 news1.html、content200.html 等类型，采用方法类似。

**业务类型的“冗余路径”**

和“伪静态页面”对应的有另外一种动态页面，这些页面按照默认的扫描规则，会被自动过滤，但是根据真实的业务场景，这些页面确实不能被过滤的，如访问 demo.testfire.net 时候在“已过滤 URL”内会显示有如下的 URL 地址，过滤原因都是“路径限制”：

http://www.Test.com/default.aspx?content=inside\_community.htm

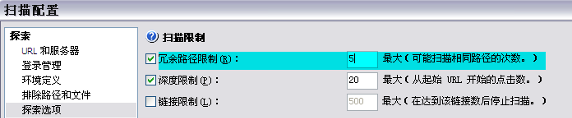
http://www.Test.com/default.aspx?content=inside\_press.htm

http://www.Test.com/default.aspx?content=inside\_executives.htm

选择 URL 地址，鼠标右键“**在浏览器中显示**”，会发现这里显示的页面内容完全不一样，和上面的“伪静态页面”正好相反，这些参数相同，参数值不同的动态页面，是真正的业务页面，是不能过滤掉；如果过滤，则会很多后续的业务页面无法发现。那这些页面为什么会被过滤了呢？按照什么样的规则被过滤掉的？

在 AppScan 中，默认情况下是有一个“**冗余路径限制**”（在“扫描配置 - 探索选型 - 冗余路径限制”），默认对于冗余的页面，最多扫描 5 次，关键的问题是，什么页面被被 Appscan 认为是冗余页面呢 ?

**图 4. 冗余路径设置**



简单说：

[点击查看代码清单](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N10210)

[关闭 [x]](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "close)

*http://www.Test.com/default.aspx?content=inside\_community.htmhttp://www.Test.com/default.aspx?content=inside\_press.htm*

Appscan 是根据“?”号来分隔的，如果 ? 号前面的内容都相同，则就被认为是冗余页面，所以上面的页面就是冗余页面了。

遇到这样情况的页面，最多被访问 5 次。而这 5 次，具体是使用了哪些参数，是随机的，具体访问到的页面也会在“应用程序数据”视图的“已访问的 URL”中查看：

[点击查看代码清单](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N1021B)

[关闭 [x]](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "close)

*http://www.Test.com/default.aspx?content=business.htmhttp://www.Test.com/default.aspx?content=business\_lending.htmhttp://www.Test.com/default.aspx?content=inside\_contact.htm*

可是，在本例中 content 参数值不同的时候，其实根据业务逻辑，不应该算作“冗余页面的”，而按照配置，也会被自动过滤了，遇到这种情况，就需要考虑增加“冗余路径限制”，如设置为 20 或者 50。以可以更多次访问这些页面。

这些情况经常存在于跳转参数等情况。

顺便备注下，“冗余路径限制”，功能设置的目的是为了处理类似论坛 BBS 等页面，只有文本内容不同，页面架构完全相同的页面：如

http://www.Test.com/showthread.php?id=1

http://www.Test.com/showthread.php?id=2

http://www.Test.com/showthread.php?id=3

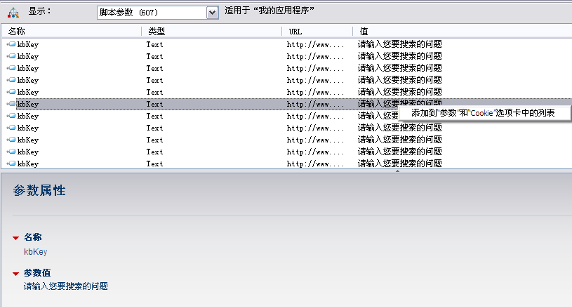
而我们在测试 demo.testfire.net 时候会发现每次的安全测试结果都可能有差别，一个很大的原因就是每次访问的页面是不同的，就是这个设置的影响。

**分析重复的“脚本参数”**

在上面的步骤中，分析了“伪静态页面”，对其应该通过“排除路径或者文件名”的方法设置排除规则；而对于“业务类型的冗余路径”，则需要通过增加“冗余路径显示”个数等的方法进行扩充，以扫描到这些 URL。我们在这个步骤来分析另外一种参数，**脚本参数**。

在“**我的应用程序数据**”树状结构下，鼠标选择目录以后，在右边视图中选择“**脚本参数**”，然后查看是否存在不同页面（URL) 存在相同或者类似参数的情况：如下图，在不同 URL 中，都存在 kbKey 参数，默认的参数值是“请输入您要搜索的问题”：

**图 5. 脚本参数**



[图 5 大图](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/origin_image005.png" \t "_blank)

访问这些 URL，发现每个页面内都包含了一个搜索功能，这就是为什么在不同页面都发现了该参数。而从业务角度，这些搜索页面在一个 URL 中进行测试以后，没有必要在另外一个页面也进行测试。而且该参数值的变化，可以认为是冗余页面，没有必要进行下一步的重新探索和测试。这可以通过上图中，选择该参数后，鼠标右键，选择“添加到‘参数和 Cookie’选项卡中的列表”来实现。选择后弹出下面的页面：

**图表 6　添加参数定义（根据参数来设置冗余路径）**



[图 6 大图](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/origin_image006.png" \t "_blank)

在该页面中，点击“其他选项－冗余调整”，取消选择任何一个选择框，则表示无论是否含有该参数，无论该参数值是否发生变化，都不认为是新页面，没有必要重新测试，而且不应该因为该参数的变化去影响其他参数的测试。

我们知道，AppScan 中的测试，是针对页面的每个参数进行的，而且一个参数值的变化会要求重新测试其他的参数，所以该设置，可以大大减少测试用例数。

关于更多的设置说明，可以参照下面的解释：

**表 3. 设置说明**

| **选框** | **选中时 ...** |
| --- | --- |
| 只要添加或除去此参数 /cookie，便再次探索 URL。 | 在探索阶段，如果两个 URL 的唯一区别在于一个包括此参数，而另一个不包括此参数，那么将其视为不同 URL，并且对两者都进行探索。 例如，如果是以下两个 URL，*两者*都将进行探索： ...page.jsp ...page.jsp?thisParam=Value 如果您取消选中此复选框，那么在此情况下，将仅发送一个请求，而其他请求将被废弃。 |
| 只要此参数 /cookie 的值更改，便再次探索 URL。 | 在探索阶段，如果两个 URL 的唯一区别在于此参数 /cookie 的值，那么将其视为不同 URL，并且对两者都进行探索。 例如，如果是以下两个 URL，*两者*都将进行探索： ...page.jsp?thisParam=Value1 ...page.jsp?thisParam=Value2 如果您取消选中此复选框，那么在此情况下，将仅发送一个请求，而其他请求将被废弃。 |
| 只要添加或除去此参数 /cookie，便重复所有相邻参数 /cookie 测试。 | 在测试阶段，如果两个 URL 的唯一区别在已添加或除去了此参数，那么将其视为不同 URL，并且再次测试相邻参数。 例如，如果是以下两个 URL，将为相邻参数生成*两组完整*的测试，每个 URL 一组。 ...page.jsp?adjacentParam=<test\_this> ...page.jsp?adjacentParam=<test\_this>&thisParam=Value 如果您取消选中此复选框，将为相邻参数仅生成一组测试。 |
| 只要此参数 /cookie 的值更改，便重复所有相邻参数 /cookie 测试。 | 在测试阶段，如果两个 URL 的唯一区别在于此参数 /cookie 的值，那么将其视为不同 URL，并且再次测试相邻参数。 例如，如果是以下两个 URL，将为相邻参数生成*两组完整*的测试，每个 URL 一组。 ...page.jsp?adjacentParam=<test\_this>&thisParam=Value1 ...page.jsp?adjacentParam=<test\_this>&thisParam=Value2 如果您取消选中此复选框，将为相邻参数仅生成一组测试。 |

**查看每个目录页面个数**

如果一个扫描任务，本身的已访问 URL 数超过 5000，评估的要运行的安全测试用例数超过 20，000，则建议进行扫描配置的分析，并根据分析结果，决定是否需要进一步的任务分解和分工。

我们在“**我的应用程序数据**”树状结构下，鼠标选择目录以后，在右边视图中选择“已访问的 URL()”，记录 URL 数目，如果该目录 URL 数目比较大（超过 500）则可以考虑为该目录单独建立一个扫描任务，只扫描该目录下面的链接。

**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**执行阶段**

根据在“计划阶段”确定的扫描策略，和进行的扫描设置，重新进行探索（扫描菜单依次选择：重新扫描 - 重新探索）；后继续分析页面数和测试用例数目，如果控制页面数 5000 个以内，测试用例数 20，000 个以内，则可以直接进行扫描；如果没有，建议继续分析，优化扫描配置。

**分阶段测试**

AppScan 的扫描过程分为“探索”和“测试”两个阶段，默认情况下，使用的是完全扫描模式，即是边探索边测试的。如果网站比较大，建议考虑先探索后测试的模式。

如当 URL 达到 5000，需要进行的测试达到 50000 的时候，可以暂停扫描，手工停止探索，选择“**继续仅测试**”。对已经发现和分析的页面进行测试，测试完毕，再来选择“**继续仅探索**”，即：

**继续仅探索 --- 继续仅测试—继续仅探索 - 仅测试的一个循环过程。**

在这个过程，一个阶段结束以后，建议查看下 .Scan 文件的大小，如果大小超过了 500M，则建议考虑任务分解，可以根据目录把一个扫描任务分解为多个，或者根据扫描策略来进行分解。

该方法是利用了 AppScan 扫描过程中，探索测试可以分离，而且支持扫描过程中断后继续扫描的特性。

**按照业务分解扫描任务**

在实际工作中，我们扫描的一个大型网站，往往包含多个频道，而每个频道可能需要的扫描配置都不同，这些配置甚至互相冲突。如一个网站的提供了 BBS 论坛功能：

http://www.Test.com/WWW.TEST.COM/showthread?channel=1&thread=1001

http://www.Test.com/WWW.TEST.COM/showthread?channel=30&thread=2001

对于这样的页面，访问后发现页面结构相同，只是文本内容不同，则应该使用“冗余路径限制”参数，控制扫描次数，没有必要多次扫描。

同时，该网站的一个服务频道存在如下的页面：

http://www.Test.com/default.aspx?content=inside\_executives.htm

http://www.Test.com/default.aspx?content=privacy.htm

即上面提到的业务类型的“冗余路径”，应该多次扫描，配置上要求增大“冗余路径限制”参数。

在这种情况下，就很有必要根据业务分别建立扫描任务，每个任务采用不同的扫描配置。

**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**检查阶段**

在扫描执行过程中，需要检查，看是否存在下面的情况：

1. 提示网络连接不上，或者提示部分页面无法打开。则检查是否是扫描速度过快，服务器不能承受不了，根据情况修改扫描配置 - 连接 - 通信和代理，增加“超时”数，并考虑减少“并发线程数”，以允许更长时间的等待页面影响并减少对服务器的访问连接数。
2. 发现扫描出的安全问题，包含我们不关心的安全隐患，则取消掉这些规则。如发现了一个安全隐患，类型是“SQL 注入文件写入（需要用户验证）”，该问题是需要用户根据提示来检查的，并且是针对 SQL 数据库的，如果我们使用的数据库不是 SQL 数据库，或用户确认后没有发现线索，则就可以在扫描配置 - 测试 - 测试策略中取消选择该策略。
3. 执行“计划阶段”的检查，看是否还存在“伪静态页面”，“业务类型的冗余路径”等，如果存在，则调整扫描配置。

**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**分析阶段**

在分析阶段，结合业务特点，检查是否扫描范围，分析扫描结果，并针对扫描出来的问题，进行分析，产生多种类型的报告等。

**扫描结果检查**

扫描结束后，建议切换到“应用程序数据”视图中，对页面进行分析，检查是否核心页面都被测试到了。重点检查如下部分：

1. 交互式 URL：一些页面，必须输入正确的信息，才可以跳转到下一个页面，比如查询手机欠费的页面，必须输入正确的 11 位手机号码；查询身份信息的页面，必须输入 18 位的身份证号才可以进入后续页面。如果没有配置，AppScan 怎么知道输入这些信息？所以如果存在“交互式”URL，可以选择该 URL 以后，鼠标右键，选择手动探索，在 AppScan 浏览器中访问这些页面，输入对应的数据，则 AppScan 会自动记录这些输入，并填充到扫描配置 - 自动表单填充中。
2. 中断链接：看哪些页面在扫描过程中，访问出错或者无法访问，如针对 time out 的页面，就可能是因为网络原因，扫描过程中没有及时响应，可以选择“重试所有中断链接”重新进行访问。

**报告分析**

我们需要对报告进行对比分析或者报告汇总合并，方法如下：

1. 增量分析：在实际工作中，经常对一个网站进行定期扫描，那么我们可以使用报告对比功能，对比两次产生的结果，检查哪些问题已经修改，哪些是新发现的安全隐患。方法是选择报告 - 增量分析。
2. 报告汇总和合并：而如果我们在执行阶段，按照业务或者目录进行了分解，最后可能需要对多份扫描结果进行合并和汇总，合并过程中重复的问题只记录一次，如扫描任务 A 和任务 B 都发现了 apply.jsp 的 ID 参数存在 XSS 安全隐患，则合并后只记录一次。报告的合并需要使用到 AppScan 企业版，其具有 AppScan 标准版的扫描功能和强大的报告汇总功能，可以产生仪表盘，报告的对比分析，趋势分析等。可以把 AppScan 标准版的报告发布到 AppScan 企业版中，方法是菜单栏中依次选择文件 - 导出 - 将结果发布到 AppScan Enterprise。

**图 7. AppScan 标准版的扫描结果发布到企业版**



**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**案例分析**

工作中遇到一个案例，使用 AppScan 扫描扫描了 3\*24 小时，扫描的 scan 文件已经达到 9G；扫描还在持续进行中，总体进度完成了 30%，可以想象扫描速度已经很缓慢，还需要多长时间才可以完成扫描？扫描完成以后如此大的结果文件是否可以成功打开和修改保存 ?

按照我的经验，如果扫描结果文件大于 1G，那就很有必要立即停止扫描，进行配置分析。我们的分析过程如下：

1. 和用户讨论，确认关心的安全问题，根据这些安全问题制定测试策略；讨论后确定选择“SQL 注入”和“跨站点脚本编制”两种类型的安全隐患。
2. 确定网站范围，被扫描应用是典型运营商门户网站，重点要扫描门户网站自身和其上面提供的“网上营业厅”服务。
3. 分析被测网站，使用 AppScan 配置了网站主页面，然后选择“仅探索”运行 20 分钟后，发现 30，000 多个页面。停止探索，开始分析页面。
4. 分析发现该网站同一个链接，存在 http、https 访问的不同情况，而且两种访问方式访问到的页面内容相同，则过滤掉 https 的请求，集中测试 http 请求。
5. 分析发现存在大量的“伪静态页面”，如：
6. http://www.Test.com//focus/satisfy/file5.html

http://www.Test.com//focus/satisfy/file6.html

在扫描配置 - 排除路径和文件中：

排除所有该类型的页面；.\*file\d+.html

增加“例外”，对该类型的页面只扫描 file1.html 和 file20.html

1. 同时，发现了 swf 文件，应该不准备扫描 Flash，所以在“排除文件类型”中，设置根据后缀名排除 swf 文件。
2. 发现

http://www.Test.com/service

目录下存在大量如下类型的页面，都是 menu 参数值不同，访问以后发现出现的是页面中有不同的超链接：

http://www.Test.com/service/Business.do?menu=Query

http://www.Test.com/service/Business.do?menu=Open

http://www.Test.com/service/Business.do?menu=Service

确认该页面是业务类型的“冗余路径”，应该全面扫描，则需要把“冗余路径设置”调整为比较大的参数，同时该频道是网上营业厅频道，也要求用户先登录。所以针对该目录建立一个单独的扫描任务，只扫描该目录和其下子目录。

1. 分析发现 index.jsp 在多个目录下出现，而且每次出现都有两种格式，即没有参数和有固定的三个参数，每次的参数值都相同。如：

[点击查看代码清单](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "N10369)

[关闭 [x]](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "close)

*http://www.Test.com//rdwd/jfmz/jifen/index.htmlhttp://www.Test.com//rdwd/jfmz/jifen/index.html?queryType=common&applyArea=010*

*&kbKey= 请输入您要搜索的问题http://www.Test.com//rdwd/txl/rdwdznyd/index.htmlhttp://www.Test.com//rdwd/txl/rdwdznyd/index.html?queryType=common&applyArea=010*

*&kbKey= 请输入您要搜索的问题*

访问上面的页面，发现内容相同，则说明是否带这三个参数不会影响探索发现更多的页面，则可以设置这三个参数每次是否出现，是否有不同值都可以认为是同一个页面。

**设置方法：扫描配置中依次选择“参数和 Cookie”来实现。然后增加 queryType，applyArea，kbKey 三个参数，均设置为“是否有参数”、“参数是否变化”不影响测试的模式。**

1. 切换到“应用程序视图”，分析“中断链接”，发现一些页面存在“范围内容超过最大容量的”的情况，在 IE 浏览器中直接访问，发现这些页面存在死循环，页面内容无限递增。则在扫描配置 - 排除路径和文件中排除这些页面。
2. 根据以上设置，建立了两个扫描任务，均扫描“SQL 注入”和“跨站点脚本编制”。重新探索后，页面总数减少到 4000 多，测试用例数减少到接近 50，000，两个扫描任务均在 8 个小时内完成。

**[回页首](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite2/" \l "ibm-pcon)**

**总结**

AppScan 作为一种自动化的扫描设置工具，我们了解其工作原理后，需要根据被测系统的业务特点和网站结构特点，优化配置，从而可以快速的针对性扫描。

常用的设置是可以利用其“探索”功能，快速得到结构，然后分析是否存在“伪静态页面”，业务上的“冗余路径”页面，“参数重复”页面等，在扫描配置中对应设置。

同时，如果设置后网站规模还是比较大，则可以根据业务分解为多个扫描任务，从而分而攻之，快速扫描，并结合企业版等工具，综合汇总分析。

**参考资料**

**学习**

* 查看本系列的第一部分：“[Rational AppScan 扫描大型网站，第 1 部分: 工作原理及技术分析](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanlargewebsite1/)”。
* “[Rational AppScan: 全面检测 Web 安全漏洞的利器](http://www.ibm.com/developerworks/cn/onlinecourse/rational/appscanintro.html)”（developerWorks，2007 年 11 月）：您的 Web 应用安全吗？仅仅是防火墙能够保证 Web 应用的安全吗？如何避免遭受黑客利用跨站点脚本攻击（Cross Site Scripting Flaws）等由 OWASP所公布的 Web 应用安全漏洞入侵您的网站？本讲座将为您一一介绍这些关于 Web 应用安全的常见问题。同时，还将为您介绍专门面向 Web 应用安全检测的自动化工具 IBM Rational AppScan。并介绍如何运用 Rational AppScan 来自动化检测 Web 应用的安全漏洞，从网站开发的起始阶段就扫除 Web 应用安全漏洞。
* “[使用 Rational AppScan 保证 Web 应用的安全性，第 1 部分: Web 安全与 Rational AppScan 入门](http://www.ibm.com/developerworks/cn/rational/r-cn-appscan1/)”（developerWorks，2007 年 12 月）：本文将从对 Web 应用现状的分析入手，通过列举常见的攻击手段，阐明 Web 应用目前面临的挑战，同时，通过对 Rational AppScan 平台的介绍，协助企业制定 Web 应用安全解决方案，为企业的 Web 应用披上盔甲。在第一部分，将介绍 Web 安全与 Rational AppScan 的入门知识。在后续的 第二部分 将介绍如何使用 Rational AppScan 应对 Web 应用攻击。
* “[使用 Rational AppScan 保证 Web 应用的安全性，第 2 部分: 使用 Rational AppScan 应对 Web 应用攻击](http://www.ibm.com/developerworks/cn/rational/r-cn-appscan2/)”（developerWorks，2007 年 12 月）：本文将针对 Web 安全的现状、根源、以及 Rational AppScan 产品的技术细节做全面的介绍，最后阐述IBM解决方案给企业带来的深层次价值。
* “[Rational AppScan 入门简介](http://www.ibm.com/developerworks/cn/offers/lp/demos/summary/appscanintro.html)”（developerWorks，2008 年 6 月）：本次演示将帮助您了解一个面向 Web 应用程序与 Web 服务的黑箱安全扫描工具——Rational AppScan。利用 AppScan，应用程序开发团队在项目交付前，可以对所开发的应用程序与服务进行安全缺陷的扫描。
* “[IBM Rational AppScan Enterprise Edition 护驾企业 Web 应用安全，第 1 部分：企业级 Web 应用安全解决方案介绍及其特点](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanent1/)”（developerWorks，2008 年 9 月）：本文分为两部分，第一部分将从 Web 应用安全解决方案的演进入手，探究解决方案的发展之路，并对文中称之为“战略方法”的企业级解决方案及其特点进行深入分析。
* “[IBM Rational AppScan Enterprise Edition 护驾企业 Web 应用安全，第 2 部分：企业级 Web 应用安全解决方案实例](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanent2/)”（developerWorks，2008 年 9 月）：本文分为两部分，第二部分中，将从开发人员、安全管理员、经理等角色出发，用实例的形式展示出使用统一的管理平台后，给企业安全管理带来的方便性、统一性和其它优势。
* “[Web 应用安全利器：Rational AppScan](http://www.ibm.com/developerworks/cn/rational/r-cn-appscanusage/)”（developerWorks，2010 年 2 月）：IBM Rational AppScan 是全面的 Web 应用安全解决方案，为 Web 及 Web 2.0 应用提供了扫描、报告和修复建议等功能，可以帮助开发者在项目起始阶段就能全面准确地发现并解决安全问题。本文从新手入门的角度，为您全面介绍了 Rational AppScan 的使用方法，帮助您了解并在实际应用中运用这一功能强大的自动化 Web 应用安全检测工具。
* 查看演示：[Rational AppScan 入门简介：Web 应用与 Web 服务安全扫描工具](http://www.ibm.com/developerworks/cn/offers/lp/demos/summary/appscanintro.html)，本次演示将帮助您了解一个面向 Web 应用程序与 Web 服务的黑箱安全扫描工具——Rational AppScan。利用 AppScan，应用程序开发团队在项目交付前，可以对所开发的应用程序与服务进行安全缺陷的扫描。
* 通过多媒体课堂“[Rational AppScan: 全面检测 Web 安全漏洞的利器](http://www.ibm.com/developerworks/cn/onlinecourse/rational/appscanintro.html)”，深入了解 Rational AppScan。
* 访问 [UML 资源中心](http://www.ibm.com/developerworks/cn/rational/uml/)，获得关于统一建模语言（Unified Modeling Language，UML）的入门知识、技术资源和最佳实践。
* 访问 [IBM developerWorks 中国网站 Rational 专区](http://www.ibm.com/developerworks/cn/rational/)，获得关于 IBM Rational 软件交付平台（Rational Software Delivery Platform）产品的技术资源和最佳实践。
* 订阅 [IBM developerWorks 时事通讯](http://www.ibm.com/developerworks/cn/newsletter/)，一份关于 developerWorks 指南、文章、下载、社区活动、网络广播和技术讲座的电子周刊。