Web应用测试用例生成研究

彭树深 顾 庆 陈道蓄
(南京大学计算机科学与技术系 南京 210093)

摘 要 随着Internet的高速发展,Web应用成为软件的主流之一,Web应用程序也变得越来越复杂,从多方面着手提高Web应用的质量是Web应用开发者的必然选择。Web应用测试是保证Web应用程序质量的重要手段,测试的核心工作是生成测试用例。把现有的Web测试用例生成方法归结为4类:Capture/Replay方法、HTML分析法、源代码分析法、User-Session分析法,随后对这4类方法进行详细讨论,并归纳出这4类方法生成测试用例的一般步骤。最后总结了这4种方法的优缺点,从方法实现的难易程度、覆盖率等方面比较了它们的性能。

关键词 Web应用测试,Web测试用例生成,Web测试用例生成分类
中图法分类号 TP311.15 **文献标识码** A

Study of Test Case Generation for Web Applications

PENG Shu-shen GU Qing CHEN Dao-xu
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)

Abstract Along with the fast evolution of Internet, Web application development has become one of the main categories of software application development, and Web application programs have become more and more complicated. Improving the quality of Web applications from different aspects is inevitable task of Web application developers. An important way of ensuring the quality of Web applications is software testing, the core work of which is test case generation. In this paper, we classified the methods for Web test case generation into four categories: Capture/Replay, HTML analysis, Source code analysis, and User-Session analysis. We discussed these four methods in detail, and concluded general steps of generating test cases in these methods. Finally, we listed the advantages and disadvantages of the four method categories, and compared their performance from various perspectives, including the effort needed to implement the method, maximum attainable code coverage, and so on.

Keywords Web application testing, Web testing case generation, Web testing case generation classification

1 引言

随着Internet的发展,Web应用渗透到了人们生活中的各个角落。Web测试是保证Web应用质量的重要手段,测试的核心是生成测试用例。

基于常用的Browser/Server模式,Web应用的一般模型如图1所示。

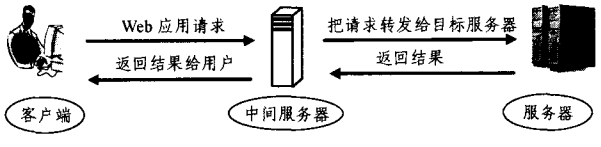

图1 Web应用模型

客户端即终端用户,中间服务器为局域网服务器、代理服务器等。服务器是提供Web应用的服务器,研究测试用例的生成应从这3处着手。

Web应用具有分布、异构、并发和平台无关等特性[2]。按照Web测试用例生成方法所采用的技术,我们认为Web测试用例的生成方法可归结为4类。

方法1:Capture/Replay方法,基于Capture/Replay技术来产生测试用例[4,5];

方法2:HTML分析法,分析客户端HTML代码,建立模型,从模型来产生测试用例[1,4,6,9-11,13,21];

方法3:源代码分析法,分析服务器端源代码来产生测试用例[3,9,22];

方法4:User-Session法,基于用户会话(User-Session)来产生测试用例[7,8,12]。

方法1主要在客户端和中间服务器处实现,方法2主要在客户端实现,方法3和方法4主要在服务器端实现。

本文结合图2所示的学生信息管理系统(SIMS, Student Information Management System),来讨论上述4类测试用例生成方法。

该应用采用JSP+Servlet实现,用户从login.jsp登录。登录成功之后,可以修改密码,查看某个学年的成绩,所有的表单请求(带箭头的实线)都发送到processor这个Servlet。

到稿日期:2009-07-02 返修日期:2009-09-21 本文受国家"863"项目(编号:2006AA01Z177)基金资助。
彭树深 男,硕士生,主要研究方向为Web测试等,E-mail:shushengpeng@121.com;顾 庆 男,教授,主要研究方向为软件测试;陈道蓄 男,教授,博士生导师,主要研究方向为分布式计算与软件工程。

图中，虚线表示页面之间存在链接，页面中的表单字段用带下划线的文字表示，fromName 是隐藏字段。welcome. jsp 是一个带 3 个框架的页面，初始时装载 top. jsp，grade. jsp 和 left. jsp。

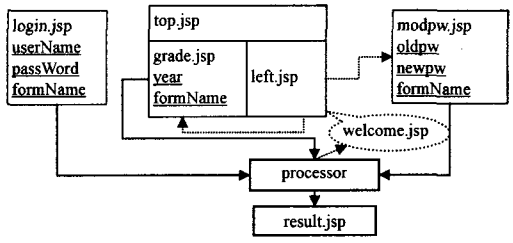

图 2　学生信息管理系统

本文第 2 节到第 5 节讨论 4 类 Web 测试用例生成方法；第 6 节对这些测试用例生成方法进行综合比较；最后讨论下一步的研究工作。

2　Capture/Replay 方法

Capture/Replay 法产生测试用例的基本流程：

Step1　测试人员打开 Web 应用，执行浏览、搜索等操作；

Step2　采用 Capture 技术记录所执行的操作和所操作的内容；

Step3　根据记录的数据生成测试用例。

Capture/Replay 技术产生测试用例，需要测试人员手动产生原始测试用例，该方法耗时、效率低、难以达到有效的覆盖率，且不易规范化。

3　HTML 分析法

HTML 分析法的基本流程如下：

Step1　选定某一页面 P_0（一般为 Web 应用首页），读取 P_0 的 HTML 代码；

Step2　从 HTML 中解析出所有可达页面，提取建模所需元素，并利用所获得的信息建立模型，如 FSM、UML 模型；

Step3　判断是否已满足终止条件，是，执行 Step4；否，选出一个可达的页面 P_1，令 $P_0::=P_1$，转 Step1；

Step4　从模型产生测试用例。

3.1　典型的 HTML 分析法

3.1.1　导航测试树

David C Kung 等人[9]从 Web 应用的行为视角，提出了导航测试树。要生成导航测试树，先要生成 Web 应用的对象关系图（ORD，Object Relation Diagram），为此把整个 Web 应用视为一个对象。对象由属性和操作组成，Web 应用中有 3 种子对象：服务端页面、客服端页面和组件；属性包括程序变量、HTML 元素等；方法包括：脚本函数和应用程序中的函数。由 ORD 图生成页面导航图，由页面导航图再生成导航测试树。

图 3 是 SIMS 用户登录形成的导航测试树，[var＝K1] url-A 表示通过 url-A 到达 processor 时所带参数集合为 K1。

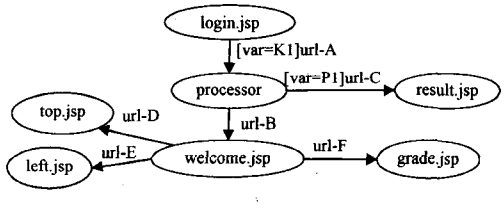

图 3　导航测试树

运用图论知识，从导航测试树可得到测试路径，进而生成测试用例。如图 3 中，从 login. jsp 到 result. jsp 测试路径有 [var＝K1]url-A，[var＝P1] url-C。

3.1.2　FSM 模型

FSM 可以完整地描述 Web 应用[1]，但容易遇到状态爆炸问题。A. A. Andraws 等人[1]提出了分层聚集 FSM，通过添加限制条件，减少无意义测试用例的生成，缓解了状态爆炸问题。该方法基本思想如下：

1）按功能自顶向下把 Web 应用划分成若干聚簇，聚簇可再划分子聚簇；

2）分析底层聚簇，找出状态和状态转换条件，建立底层的 FSM；

3）逐级分析高层聚簇，把底层的 FSM 聚合成高层的 FSM；

4）最后得到描述整个 Web 应用的应用级 FSM，从应用级 FSM 产生测试用例。

简单起见，每个 FSM 都是单入口、单出口。如果有多个入口节点，则添加一个虚拟的入口节点，指向各个入口节点，其状态转换条件为空；如果有多个出口节点，则添加一个虚拟的出口节点，各个出口节点都指向该节点，其状态转换条件为空。

图 4 是 SIMS 按功能自顶向下的聚簇划分图，off 代表用户离开该系统，是虚拟的出口节点；图 4 中只划分了“修改密码”这一子聚簇。在后面的讨论中，用括号中的标注来代表该节点。

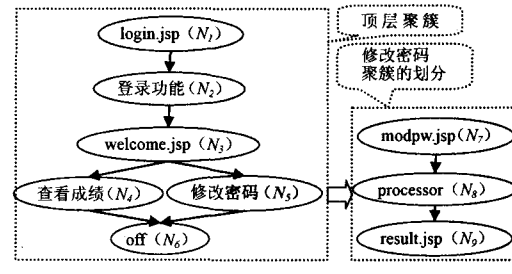

图 4　聚簇的划分

通过自底向上的分析，分析修改密码这一功能模块得到表 1 所列的结果。

表 1　为“修改密码”添加的信息

转换	限制条件	Σ	Ω
(N_7，N_8)	Required(oldpw，newpw，URL，button) S(A(oldpw，newpw，URL)，button)	S_1	N_8
(N_8，N_9)	Required(login_result，URL)	S_2	N_9

“限制条件”列举了在此次转换中必须满足的条件；Required 表示在此次状态转换中，括号里面变量的值不能为空；S 表示括号里面的变量必须按顺序提交，A 代表括号里面的变量可以无序；Σ 列把每个“限制条件”编码成一个输入字母表；Ω 列表示输出或者发送模式；例如转换（N_7，N_8），表示状态从 N_7 装换到 N_8 时必须有旧密码（oldpw）、新密码（newpw）、URL、提交按钮（button），而且 button 必须在所有值之后，输出为 N_8。

同理可得“顶层聚簇”的结果，见表 2，Required（N_2）表示限制条件要加上节点 N_2 中的限制条件。continue-use（userName，passWord）表示在此次转换的时候必须传递 user-

Name,passWord 的值。这两个值在后面还会用到。

表 2　为"顶层聚簇"添加的信息

转换	限制条件	Σ	Ω
(N_1,N_2)	Required(userName,passWord,formName,URL,button), S(A(userName,passWord,formName,URL),button), Required(N_2),continue-use(userName,passWord)	S_3	N_2
(N_2,N_3)	Required(URL),continue-use(userName,passWord)	S_4	N_3
(N_3,N_4)	Required(N_4),continue-use(userName)	S_5	N_4
(N_3,N_5)	Required(N_5),continue-use(userName,passWord)	S_6	N_5
(N_4,N_6)	Required(logout)	S_7	N_6
(N_5,N_6)	Required(logout)	S_8	N_6

把 Σ 作为输入，Ω 作为输出，结合"限制条件"和聚簇划分图，很容易就能生成 FSM。

分析 FSM 可得到测试序列，例如覆盖 SIMS 的序列为：

1) $N_1;S_3;N_2;S_4;N_3;S_5;N_4;S_7;N_6$

2) $N_1;S_3;N_2;S_4;N_3;S_6;N_5;S_8;N_6$

对节点为聚簇的节点可进一步展开；如对节点 N_5 做如下展开 $N_5:N_7;S_1;N_8;S_2;N_9$，则覆盖 SIMS 的序列变为：

1) $N_1;S_3;N_2;S_4;N_3;S_5;N_4;S_7;N_6$

2) $N_1;S_3;N_2;S_4;N_3;S_6;N_7;S_1;N_8;S_2;N_9;S_8;N_6$

3.1.3　UML 模型

Filippo Ricca 等人采用 UML 模型来描述 Web 应用[13]，从 UML 模型可以很方便地生成 Web 测试用例。该方法首先建立 UML 元模型；随后读取 Web 应用的 HTML 代码，分析代码，把 Web 应用用 UML 模型描述；最后按照需要缩减 UML 图，从 UML 图生成测试用例。

描述 Web 应用的 UML 元模型如图 5 所示。

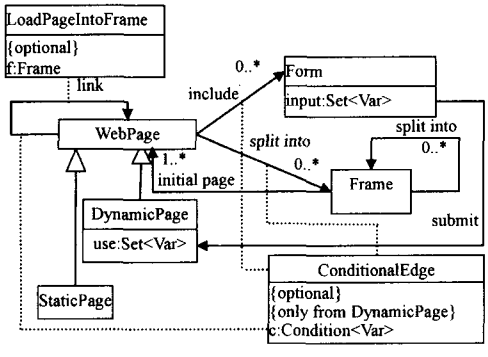

图 5　Web 应用结构的 UML 模型

读取网页的 HTML 代码提取表单、表单字段、超链接等，把 Web 应用用 UML 模型表示。图 6 是用 UML 模型表示 SIMS 中的用户登录。

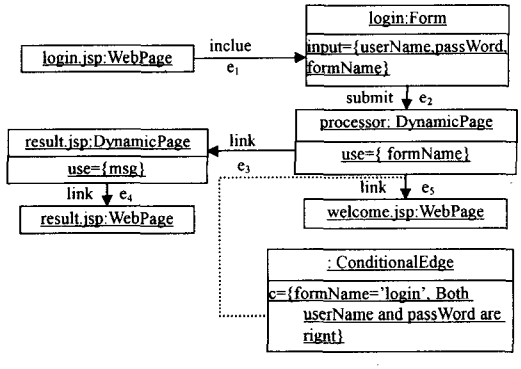

图 6　Web 应用 UML 模型的实例

按需对 UML 图进行缩减后，通过计算路径表达式即可生成测试用例，路径表达式由变量(如图 6 中的 e_1，e_2)，"("，"+"，"*"3 个操作符构成。图 6 的路径表达式为 $e_1e_2(e_3e_4+e_5)$；如果从 result.jsp 添加一个链接，指向 login.jsp，该边记为 e_6，则图 6 的路径表达式为 $(e_1e_2e_3e_4e_6)^*+(e_1e_2e_3e_4e_6)^*e_1e_2e_5$，从这个路径表达式可生成如下测试用例：$e_1e_2e_3e_4e_6$，$e_1e_2e_5$，$e_1e_2e_3e_4e_6e_1e_2e_5$ 等。

4　源代码分析法

现代 Web 应用中很多接口都是隐藏的，如 Cookie，Session，它们不像表单元素展现在客户端页面上。仅分析 HTML 代码无法发现这些隐藏接口。研究服务器端 Web 应用的源代码，可发现这些接口。

源代码分析的基本流程如下：

Step1　读取 Web 应用程序源代码；

Step2　分析源代码，得到程序的控制流图、数据流图等，并往图中的节点添加信息，如函数的调用关系[9]；

Step3　分析得到的图，产生测试用例。

William G J Halfond 等人提出了一种通过分析 Web 应用程序源代码产生 Web 测试用例的方法[3]，该方法分成两阶段实现。

第一阶段：发现参数(IP，Input Parameters)的域信息；第二阶段：识别这些参数，并按照接口对参数进行分组。接口的定义如下：

interface = IP*

IP = name,domain_information

name = ⟨string⟩

domain_information = domain_type,relevant_value*

domain_type = ANY|NUMERIC

relevant_value = ⟨string⟩ | ⟨number⟩

IP 可以是表单中的字段、Cookies、Session 或者状态变量。

如何得到 IP 信息？Web 应用会通过特定的 PF(Parameter Function)函数来访问 IP，跟踪该函数就可得到 IP 信息。

如何得到 IP 域信息？分析在 IP 值上进行的操作就可以确定域的信息，如：在 IP 的 relevant_value 上执行 Integer.parseInt()操作，那么认为这个 IP 的 domain_type 为 NUMERIC。

下面以 Java 为例，说明该方法的实现过程。在 Java 中，我们把 request.getParameter() 视为一个 PF。图 7 是 SIMS 中 Servlet processor 的部分代码。

```
...
1.  String formName = (String) request.getParameter("formName");
2.  String validUserName = "admin";
3.  int validPassWord = 123;
4.    if(formName.equals("login")){
5.      String userName = (String) request.getParameter("userName");
6.    int passWord = Integer.parseInf((String)request.getParameter("year"))
7.      if(! userName.equals(validUserName) || passWord ! = validPassWord) {
8.    String msg = "错误的用户名或密码!";
```

```
9.    request. setAttribute("msg",msg);
10.   RequestDispatcher redirect = request. getRequestDispatcher
      ("infor/result. jsp");
11.   redirect. forward(request,response);
12.   }
13.   else
      response. sendRedirect("infor/welcome. jsp");
14.   }
15.   else if(formName. equals("viewGrade")){…}
16.   else if(formName. equals("modpw")){…}
      …
```

图 7　processor 的部分源代码

分析源代码可得到 3 个接口：login，viewGrade 和 modpw，其中 login 接口可表示为 interface = {'formName'，'userName'，'passWord'}。

formName 的域类型为 ANY，有 3 个相关值：login，viewGrade，modpw；userName 的域类型为 ANY，相关值为"admin"。passWord 的域类型为 NUMERIC，相关值为 123。根据这个接口的信息就能产生测试用例，如：

http://localhost: 8080/StudentInfo/processor? formName = login&userName=admin&passWord=123

单个 Web 应用可能由多种编程语言实现，分析源代码来产生测试用例非常困难。

5　User-Session 分析法

用户向 Web 服务器提出 HTTP 请求时，服务器会把当前的请求记录在日志文件中，对 Web 服务器的配置文件稍加修改，就可得到更详尽的日志文件[15]。

从访问日志生成 Web 测试用例，代价低，生成的测试用例能真实反映 Web 应用的使用情况。

User-Session 分析法的基本流程如下：

Step1　修改 Web 服务器的配置文件，记录测试所关心的数据；

Step2　挖掘访问日志，提取用户所关心的数据；

Step3　重组数据，生成测试用例。

Sebastian Elbaum 等人于 2003 年提出了通过挖掘 Web 访问日志来产生测试用例的方法[7]。其基本思想如下：

首先定义用户会话：从第 i 个用户首次访问 Web 应用开始，到该用户退出网站或者用户两次访问之间的时间间隔大于某一上限。在这过程中，用户执行的所有操作构成一个用户会话，记为 U_i。

U_i 由若干访问请求 r 组成，即 $U_i=(r_{i1},\cdots,r_{im})$。对 r_{ij}，r_{ik}，若 $j<k$，则请求 r_{ij} 先于 r_{ik}，所有的用户会话构成一个会话池。访问请求 r 由 URL 和 0 个或多个"(参数，参数值)"对组成，如"/infor/result. jsp? msg=welcome"中，"/infor/result. jsp"为 URL，(msg，welcome)为一个"(参数，参数值)"对。

产生测试用例最简单的方法是把每个会话视为一个测试用例，按照会话的时间顺序进行测试。该方法最大的缺点是没有考虑会话之间的相互影响，所以 Elbaum 等人提出了改进方法，找出不同会话中具有相同 URL 的请求 r，让一个测试用例中包含多个会话请求。Sara Sprenkle 等学者提出了 3 种改进方法[8]。

方法 1：固定时间间隔法。设置一个阈值 t，把访问日志中同一时间间隔 t 里面的活动划分到一个测试用例；

方法 2：服务器无活动时间间隔法。设置一个阈值 t，如果服务器在超过 t 的时间间隔内无任何活动，则以此为分界点，前后属于不同的测试用例；

方法 3：带参数的用户会话。扫描整个日志，针对某个会话，找出该会话最早活动时间 f 和最后活动时间 l。把日志中 f 和 l 之间所有用户的所有活动视为一个测试用例，这样测试用例数目等于用户会话数目。其突出好处是考虑了用户之间的交叉影响，坏处是测试起来非常地费时。

Jessica Sant 等人[12]提出了通过统计分析技术和机器学习技术来挖掘 Web 日志，产生测试用例。

6　4 种测试用例生成方法的比较

表 3 为 4 类方法的综合比较。

表 3　4 类方法的优缺点比较

方法 \ 优缺点	优点	缺点
Capture/Replay 法	1. 原理简单易行； 2. 与 Web 应用平台的实现技术无关，即与 Web 应用平台所使用的操作系统、数据库、Web 服务器和编程语言等无关。	1. 要耗费大量的人力劳动来产生测试用例，费时；
HTML 分析法	1. 能对 Web 应用进行多方面的测试，如功能测试、安全测试等； 2. 与 Web 应用平台的实现技术无关； 3. 便于在客户端完成验收测试； 4. 便于使用测试领域已有的研究成果。	1. 无非常有效的方法自动填充页面中的表单项； 2. 页面中有很多链接时，如何选择下一个链接进行访问需要进一步研究； 3. 从模型生成测试用例时，容易产生组合爆炸； 4. 在搜索模型时，容易遇到环路问题。
源代码分析法	1. 能很好地发现 Web 应用程序的隐藏接口，让测试更加充分； 2. 能获得较高的代码覆盖率。	1. 通用性不强，对不同的 Web 编程语言需要建立不同的模型。
US 法	1. 生成测试用例的原始数据很容易获得； 2. 数据能够真实反映实际使用情况； 3. 非常适合回归测试； 4. 与 Web 应用平台的实现技术无关。	1. 很难检测到 POST 方式传递过来的参数值。 2. 需要有访问数据积累，在应用投入使用之前，无法运用此方法进行测试； 3. 很难测试到从未使用过的功能。

为量化比较各类方法，以 HTML 分析法为基准，即认为 HTML 分析法的所有性能居中，从方法实现、测试用例生成、覆盖率、安全测试能力、通用性等角度对 4 类方法做比较，见表 4。其中测试范围指测试方法是否能对客户端、服务器端等多方面做测试。

表 4　4 种方法的性能比较

性能参数 \ 方法	C/P 法	HTML 分析法	源代码分析法	US 法
方法实现	易	中	难	易
重用原有测试用例	难	中	中	易
生成新的测试用例	易	中	易	易
最大覆盖率	低	中	高	中
安全性测试能力	低	中	高	中
平台通用性	中	中	低	中
测试范围	低	中	低	低

结束语　本文首先将现有的 Web 应用测试用例生成方法归结为 4 类；随后对这 4 类方法中典型的方法做了归纳介绍，并且给出了这 4 种方法生成测试用例的一般步骤；最后对

4类方法的优缺点进行了总结，并从多方面对它们的性能进行了比较。

Web应用的测试用例生成已取得了不少成果，但是与实际应用的需求还有很大的差距。在自动化测试时，如何填充表单项仍是需要解决的难点；Web2.0出现后，Ajax和RSS的测试对Web测试者提出了新的挑战。由于Web应用越来越多地融入社会生活，人们很多重要的活动都通过Web应用来实现，如电子商务、电子政务、安全性测试也会成为大家关注的研究热点。

参考文献

[1] Andrews A A, Offutt J, Alexander R T. Testing Web Applications by Modeling with FSMs[J]. Software Systems and Modeling, 2005, 4(2): 326-345

[2] 许蕾，徐宝文，陈振强. Web测试综述[J]. 计算机科学，2003，30(3)：100-104

[3] William G J H, Alessandr O. Improving Test Case Generation for Web Applications Using Automated Interface Discovery[C]// Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. New York, 2007: 145-154

[4] Xu Lei, Xu Baowen. Testing Forms in Web Applications Automatically[J]. Wuhan University of Natural Sciences, 2006, 11(3): 561-566

[5] Rational robot[OL]. http://www.rational.com/products/robot/index.jsp

[6] Jia X, Liu H. Rigorous and Automatic Testing of Web Applications[C]// 6th IASTED International Conference on Software Engineering and Applications (CTIRS). San Francisco, 2002: 280-285

[7] Elbaum S, Karre S, Rothermel G. Improving Web Application Testing with User Session Data[C]// International Conference on Software Engineering(ICSE2003). Portland, 2003: 49-59

[8] Sprenkle P S, Gibson P E, Sampath P S, et al. A case study of automatically creating test suites from web application field data [C]//Proceedings of the 2006 Workshop on Testing, Analysis, and Verification of Web Services and Applications. 2006: 1-9

[9] Kung D C, Liu C H, Hsia P. An Object-Oriented Web Test Model for Testing Web Applications[C]// Proceedings First Asia-Pacific Conference on Quality Software (COMPSAC2000). Taipei, Taiwan, China, 2000: 111-120

[10] Benedikt M, Freire J, Rice P, et al. VeriWeb: Automatically Testing Dynamic Web Sites[C]// Proceedings of 11st International WWW Conference. Honulolu, 2002

[11] 黄陇，李诺，金茂忠，等. 基于DataPool的Web测试数据生成与维护方法[J]. 计算机科学，2006，33(10)：272-274

[12] Sant J, Souter A, Greenwald L. An Exploration of Statistical Models for Automated Test Case Generation[C]// Proceedings of the Third International Workshop. St. Louis, Missouri, 2005: 1-7

[13] Ricca F, Tonella P. Analysis and Testing of Web Applications [C] // International Conference on Software Engineering (ICSE2001). Toronto, Ontario, Canada, 2001: 25-34

[14] Halfond W G, Orso A. AMNESIA: Analysis and Monitoring for NEutralizing SQL-Injection Attacks[C] // Proceedings of the IEEE and ACM International Conference on Automated Software Engineering (ASE 2005). Long Beach, CA, USA, 2005: 174-183

[15] Coar K, Bowen R. Apache Cookbook[M]. Sebastopol: O'Reilly Media, Inc, 2003

[16] Apache tomcat[OL]. http://tomcat.apache.org/

[17] 邓小鹏，邢春晓，蔡莲红. Web应用测试技术进展[J]. 计算机研究与发展，2007，44(8)：1273-1283

[18] Ash L, et al. The Web Testing Companion: The Insider's Guide to Efficient and Effective Tests[M]. New Jersey: John Wiley & Sons, Inc, 2003

[19] Nguyen H Q, Johnson B, Hackett M. Web application testing: test planning for mobile and internet-based systems[M]. New Jersey: John Wiley & Sons, Inc, 2003

[20] Powell T A, et al. HTML: The Complete Reference[M]. Columbus: McGraw-Hill Education, 2002

[21] Liu C, Kung D, Hsia P. Object-based Data Flow Testing of Web Application[C]// The First Asia-Pacific Conference on Quality Software. Hong Kong, China, 2000: 7-16

[22] Wassermann G, Su Z. Static detection of cross-site scripting vulnerabilities[C]// ICSE. 2008: 171-180

(上接第45页)

攻击代价的网络抗攻击性能评估指标体系，依据评估样本值之间的相似性和差异性提出了遗传投影寻踪评估模型，采用拒绝服务类攻击方法对被测网络系统进行抗攻击性能评估来验证了算法的有效性。投影寻踪评估算法既充分发挥了投影寻踪处理高维数据的突出优势，又避免了层次分析法等评估方法对人为确定参数或评价指标的不足。

在构造投影指标函数之前，对评估样本数据进行一致无量纲化处理以消除各个评估指标的量纲效应，如何使数据的无量纲化处理更加合理是一个努力的重点。

参考文献

[1] Wack J, Tracey M. Guideline on Network Security Testing [Z]. National Institute of Standards and Technology, 2002

[2] Herzog P. Open Source Security Testing Methodology Manual 2.0 [Z]. http://isecom.securentled.com

[3] 侯一凡. 抗攻击能力评价指标体系的构建与评价方法研究[D]. 郑州：解放军信息工程大学，2007

[4] 鲜明，包卫东，等. 网络攻击效果评估导论[M]. 长沙：国防科技大学出版，2007

[5] 王会梅，江亮，鲜明，等. 计算机网络攻击效果灰色评估模型和算法[J]. 通信学报，2009，30(11A)：17-22

[6] 金菊良，魏一鸣. 复杂系统广义智能评价方法与应用[M]. 北京：科学出版社，2008

[7] 周颖，王雪松，徐振海，等. 雷达电子战效果及效能评估的一般性思考[J]. 系统工程与电子技术，2004，26(5)：617-620

[8] 汪立东. 操作系统安全评估与审计增强[D]. 哈尔滨：哈尔滨工业大学，2002