基于EAI和AOP的软件安全测试及应用研究

蒋廷耀 王训宇 马 凯 关国翔

(三峡大学电气信息学院 宜昌 443002)

摘 要 提高软件安全测试方法的易操作性和可复用性在软件工程中具有重要的研究意义。研究了一种用于测试软件安全的基于EAI模型(Environment Application Interaction Model)的软件错误注入测试方法,提出了基于面向方面编程(AOP)实现软件错误注入的安全测试模型(AOEAI)及其实现方法。对应用软件进行了实际测试,实验结果表明,该方法不仅是有效的,而且具有易操作性和可复用性强的特点。

关键词 EAI模型,面向方面编程,软件安全测试,SQL注入攻击

中图法分类号 TP312 **文献标识码** A

Research and Application of Software Security Testing Based on EAI and AOP

JIANG Ting-yao WANG Xun-yu MA Kai GUAN Guo-xiang

(College of Electrical Engineering & Information Science, China Three Gorges University, Yichang 443000, China)

Abstract Improving the easy operability and reusability of the software security testing method is important in software engineering. After introducing the security testing method of fault injection software testing based on EAI model, this paper presented a security testing model called as AOEAI and a method to carry out fault injection based on AOP. The testing results in application programs show that this method is not only effective, but also has the characteristics of easy operability and reusability.

Keywords EAI model, AOP, Software security testing, SQL injection attacks

1 引言

对软件进行安全测试是软件测试的重要研究内容[1]。目前,进行安全测试的方法主要分为静态与动态两大类。静态测试是不执行程序代码而寻找程序代码中可能存在的缺陷或评估程序代码的过程。动态测试通过在抽样测试数据上运行程序来检验程序的动态行为和运行结果以发现缺陷。普度大学WenLiang Du等人提出的基于EAI(Environment-Application Interaction)模型的软件错误注入测试是一种白盒动态测试方法,该方法不仅能有效触发环境异常,且为何时模拟环境错误提供了系统的方法[2]。然而,WenLiang Du等人并没有深入研究如何实现注入环境错误。国内已有学者提出用包裹函数实现EAI模型的错误注入[3],但其并没有研究基于包裹函数实现方法的易操作性和可复用性。为了提高错误注入方法的易操作性和可复用性,本文提出基于EAI和AOP的软件安全测试模型——AOEAI模型,该模型采用AOP来实现EAI模型的错误注入。

2 基于EAI模型的软件错误注入测试

在EAI模型中,"系统"由"应用程序"和"运行环境"组成,所有被认为不属于运行程序的代码就属于环境,例如:用户输入、应用程序使用的环境中声明的全局变量、文件和网络等公共资源。程序员总是假定认为他们的程序会在正常的环境中运行。当这些假设成立时,程序当然是正确运行的。但是,由于作为共享资源的环境常常被其它主体所影响,尤其是恶意的用户,这样,程序员的假设就可能是不正确的。在激发安全缺陷的过程中,环境起了重要作用,程序是否能够容忍环境中的错误是影响程序健壮性的一个关键问题。

错误注入方法通过选择一个适当的错误模型,试图触发程序中包含的安全漏洞,基于EAI模型的软件错误注入方式分为间接环境错误注入法和直接环境错误注入法。间接环境错误注入法是当应用程序实体向环境实体请求输入时,通过向环境实体注入错误,实现向应用程序注入错误的目的。直接环境错误注入法是当应用程序访问环境实体(如建立、修改、读、写文件)时,通过向环境实体注入错误,实现向应用程序注入错误的目的。

3 AOP的核心思想

所谓的Aspect,从抽象意义上讲,是对系统组件的性能和语法产生一定影响的一些属性;从设计上讲,是横切系统的一些软件系统关注点;从实现上讲,Aspect是一种程序结构构造单元,它支持将横切系统的关注点封装到单独的模块单元中。面向Aspect的程序设计是一种关注点分离技术,通过运用Aspect这种程序设计单元,允许开发者使用结构化的设

到稿日期:2008-10-10

蒋廷耀(1969—),男,博士,教授,主要研究方向为软件工程、可信计算,E-mail:jiangty@ctgu.edu.cn;王训宇(1983—),男,硕士生,主要研究方向为AOP、WebGIS;马 凯(1980—),男,硕士,讲师,主要研究方向为计算机应用;关国翔(1983—),男,硕士生,主要研究方向为软件工程、配电GIS。

计和代码，反映其对系统的认识方式。AOP 的核心思想就是从主关注点中分离出横切关注点，并且可以定义一个新的程序构造“方面”来支持“横切关注点”的模块化、局部化[4]。

4 AOEAI 模型

研究发现 AOP 的“切入点-通知”(Pointcut-Advice)机制能有效地运用于软件安全测试中[5]。在此基础上，本文提出基于 EAI 和 AOP 的软件安全测试模型，即采用 AOP 方法实现 EAI 模型的环境错误注入。安全问题在系统运行至始至终都存在，所以说安全关注点是一个横切关注点。错误注入点不仅难找，注入时间难以把握，而且不慎会与系统安全无关的程序造成混乱。采用 AOP 的“切入点-通知”机制能将安全测试关注点分离出来，并封装成横切关注点。进一步，还可以抽象化横切关注点，以达到复用的目的。

4.1 模型定义

本文所讨论的错误注入只限于环境错误对应用程序的影响。提出的基于 EAI 和 AOP 的软件安全测试模型——AOEAI 模型，如图 1 所示。

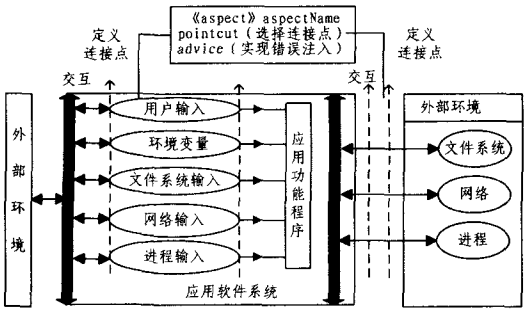

图 1 AOEAI 软件安全测试模型图

AOEAI 由一个五元组组成：

$AOEAI = (FL, FT, FIP, FK, FD)$。其中，各元组的意义如下：

(1)错误注入位置(Fault Location，FL)。如果系统需要从环境输入，则同时采用间接环境错误注入法和直接环境错误注入法；如果系统无需从环境输入，则只采用直接环境错误注入法。间接环境错误注入法是在环境和应用系统交互之后注入错误，而直接环境错误注入法是在环境和应用系统交互之前注入错误。

(2)错误触发方式(Fault Trigger，FT)。错误触发基于 AOP 的“切入点-通知”机制实现，在环境与系统交互处(即错误注入位置)定义连接点(join point)，然后通过切入点(pointcut)选择连接点。连接点是程序控制流中的事件，它可以是方法(包括构造函数)的调用或执行、对一个对象属性的访问或修改、类和对象的初始化、异常处理的执行等。切入点扮演了过滤器的角色，匹配符合其定义的连接点，而阻塞所有其他的连接点。通过这种方式，可触发特定的某种错误。

(3)错误注入方式(Fault Injections Pattern，FIP)。错误注入也是由 AOP 的“切入点-通知”机制实现的。切入点起选择连接点的作用，而通知则指定在这些连接点处的动作。在 aspect 中，将“环境错误”以“动作”的形式注入到系统中，实现环境错误注入。

(4)错误类型(Fault Kind，FK)。EAI 模型的错误类型如图 1 所示，可能出现错误的内部实体是用户输入实体、环境变量实体、文件输入实体、网络输入实体、进程输入实体；可能出错的环境实体是文件系统、网络、进程。

(5)错误注入信息数据集(Fault Data，FD)。故障注入信息数据集保存注入地址、地址上的注入前数据信息、注入后数据信息、错误注入结果。

AOEAI 模型的具体实现过程分为以下几个步骤：

1. 选定要注入的错误类型(FK)。

2. 确定错误注入位置(FL)。

3. 采用 AOP 的“切入点-通知”机制实现错误触发方式(FT)和错误注入方式(FIP)。

4. 在错误注入过程中，保存错误注入信息数据集(FD)。

5. 选定另一种错误类型，重新由步骤 2 到步骤 4，直到所有要注入的错误类型都完成。

4.2 模型分析

(1)AOEAI 软件安全测试模型的错误注入方法具有易操作性。由于安全问题是个横切关注点，按照常规的错误注入方法容易造成混乱。基于 AOP 的“切入点-通知”机制实现环境错误注入的方法简单明了，选择错误注入点的 AOP 切入点和注入错误的 AOP 通知都与原程序分离，不会产生原程序与安全测试代码的混乱。

(2)AOEAI 软件安全测试模型的错误注入方法复用性强。安全测试关注点的分离，不仅使该方法简单明了，而且有助于提高安全测试代码的复用性，针对某种错误类型可以设计一个抽象方面来提高复用性。比如对于某个系统的环境变量错误类型设计一个抽象的安全测试方面，在这个方面内部定义抽象的切入点。这样，通过相关 AOP 技术的支持，我们的测试代码不仅能测试这个系统可能多处出现环境变量的错误，而且还能用于测试其他系统可能出现的环境变量的错误。

文献[3]提出用包裹函数实现 EAI 模型的错误注入。包裹函数是添加了额外代码的函数，在其中调用了基本库函数。在程序中使用包裹函数的主要目的有二：一是增强基本函数的功能；二是处理基本函数的错误。与 AOEAI 软件安全测试模型的错误注入方法相比，采用包裹函数不具有易操作性和可复用性。

5 模型实验

5.1 SQL 注入攻击的安全测试

由于 AOEAI 模型是在 WenLiang Du 等人提出的 EAI 模型的基础上设计的，本文并不细述模型中提到的所有可能错误类型(Fault Kind)的安全测试的实现，而只选择“用户输入实体”作为一个实验，着重强调基于 AOEAI 模型实现方法的易操作性和可复用性。

在 Web 管理信息系统中，用户登录时输入用户名和密码是一种典型的用户输入，而在用户输入的过程中，系统遭受 SQL 注入攻击的现象非常普遍。SQL 注入攻击就其本质而言，是攻击者能够利用应用程序没有对用户输入的数据进行严格约束和合法性检查等漏洞，在程序中插入并执行自己构造的 SQL 语句[6]。SQL 注入攻击具有广泛性、技术难度不高和危害性大等特点，常见的 SQL 注入攻击有单引号攻击、注释符攻击、Union 注入、select 注入、Delete 注入、Update 注入等[7]。虽然在客户端可以对恶意 sql 语句进行过滤或去

除，但这并不是明智的方法，因为攻击者可以修改验证程序或者绕过客户端进行攻击，所以必须在服务器端进行安全设计。按照 AOEAI 模型的安全测试方法，本文下面将对 Web 管理信息系统的服务器端进行关于 SQL 注入攻击的白盒动态安全测试。

测试的 Web 管理信息系统采用 J2EE 构架，并基于 Hibernate 2 技术实现数据的持久化。数据库是 SqlServer 2000 SP4，AOP 实现工具是 AJDT[8]。系统中的一个 Action 类 LoginCheckAction 用于得到用户输入的用户名和密码，并进行登录验证。LoginCheckAction 的 getUserByLogin 方法根据用户名和密码，查询数据库是否存在该用户，并返回一个用户(User)列表。getUserByLogin 方法的实现代码如下所示：

```
List(User) getUserByLogin(String userName,String userPassWord) {
    try{
    String queryString="from User y where y. dengluming='"+userName+"'
and y. mima='"+userPassWord+"'";
    List(User) result = getSession(). createQuery(queryString). list();
    return result;}
    catch(HibernateException ex){
    System. out. println(ex. getMessage());
        return null;
    }
}
```

如图 2 所示，对于 sql 注入攻击的安全测试设计了一个抽象方面 SqlInject，该方面定义一个抽象的切入点，并且封装一些常见的 sql 注入攻击方法。建立抽象方面的目的在于复用，它能用于多个系统的 sql 安全测试。LoginSqlInject 方面继承 SqlInject，针对某个系统具体定义切入点。LoginSqlInject 方面可以使用其父类中封装的 sql 注入攻击方法，也可以自定义 sql 攻击方法。在本实验中，getUserByLogin 方法是安全测试的错误注入位置，在此定义连接点，并建立通知(around advice)改变 getUserByLogin 方法的参数——登录名和密码，达到绕过客户端注入 SQL 攻击的目的。

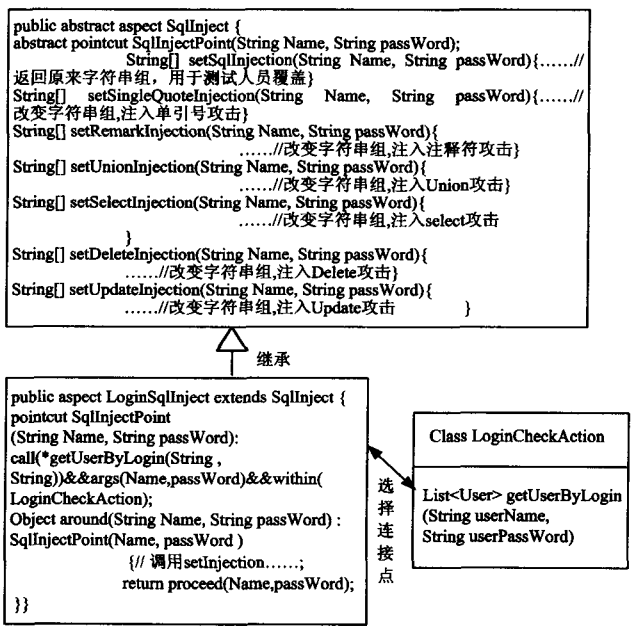

图 2 sql 注入安全测试原理图

5.2 实验结果

对 Web 管理信息系统的 sql 安全测试结果如表 1 所列。表中没有列出的注入地址是 LoginCheckAction 类的 getUserByLogin 方法，注入前数据信息是(name：myname；PW：mypw)。

表 1 Web 管理信息系统的 sql 安全测试结果

Sql 注入类型	注入后数据信息	注入错误的结果
单引号攻击	Name:myname' or '1'='1 PW:mypw	绕过系统验证，进入系统
注释符攻击	Name:myname'--PW:mypw	系统抛出异常：SQLException
Union 注入攻击	Name:myname' union from User u where u. dengluming='200610303014 ; PW:mypw	系统抛出异常：SQLException
Select 注入攻击	Name:myname'; from User u where u. dengluming='200610303014 ;PW:mypw	系统提示语法错误，对系统无影响
Delete 注入攻击	Name:myname'; delete User u where u. dengluming='200610303014 ;PW:mypw	系统提示语法错误，对系统无影响
Update 注入攻击	Name:myname'; update User u where u. dengluming='200610303014 ;PW:mypw	系统提示语法错误，对系统无影响

从表 1 中我们可以得出：基于 AOEAI 模型的软件错误注入测试方法能有效测试出系统存在 sql 注入的安全漏洞。针对该 Web 管理信息系统存在的 sql 注入的安全漏洞，在系统后期维护和重构过程中，除了加强客户端的过滤和验证以外，我们优化了数据库的安全设计，并对简单的查询使用 Hibernate 的 Criteria 进行查询，对复杂查询语句使用参数绑定。

结束语 由于软件系统开发的时间紧迫和软件测试自动化程度偏低等原因，提高软件安全测试方法的易操作性和可复用性有重要的研究意义。本文提出的 AOEAI 模型是一种基于 EAI 和 AOP 的软件安全测试模型。分析和实验结果表明该模型具有易操作性和可复用性强的特点，为软件安全测试提供了一种新的途径。

参 考 文 献

[1] 单锦辉，姜瑛，孙萍. 软件测试研究进展[J]. 北京大学学报：自然科学版，2005，41(1)：134-145

[2] Du Wenliang，Mathur A P. Testing for software vul2 nerability using environment perturbation [J]. Quality and Reliability Engineering International，2002，18 (3)：261-272

[3] 曾凡平. 一种基于 EAI 模型的软件安全测试方法[J]. 华中科技大学学报：自然科学版，2005，33：304-305

[4] Kiczales G，Lamping J，Menhdhekar A，et al. Aspect-Oriented Programming [J]// Proceedings of ECOOP'97，Lecture Notes in Computer Science. Vol. 1241，Springer ，1997：220-242

[5] Belblidia N，Debbabi M，Hanna A. AOP Extension for Security Testing of Programs[J]// Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering，CCECE'2006. May 2006：647-650

[6] Litchfield D，Anley C. The database hacker's handbook [M]. Wiley Publishing Inc，2005

[7] Anley C. Advanced SQL injection in SQL server applications [EB/OL]. http://www. creangel. com/ papers /advanced_sql_injection. pdf，An NGS Software Insight Security Research (NISR) Publication，2002

[8] AspectJ Web Site. http://www. eclipse. org/aspectj