基于模糊测试的软件安全性测试框架的研究与设计

苑立娟[1]，张育玉[2]

(1. 保定学院 信息技术系，河北 保定 071000；2. 保定学院 组织部，河北 保定 071000)

摘 要:在一个软件还未形成之前,测试工作已经介入,在这之前安全性问题是要考虑的重要元素之一。安全性测试在实物软件开发之前应该提前规划并开始实施。为了通过软件测试找出安全隐患,本文针对软件隐含的安全漏洞进行深入研究,将软件的安全漏洞测试工作分为安全性测试规划、伴随测试、整合测试 3 个阶段,概念性和动作性 2 个区域。模糊测试在测试中效率较高,适合安全漏洞测试,结合实践测试,提出基于模糊测试的软件安全性测试架构,在软件从无到有的整个过程中找出潜在的安全性问题,进而可以有效提升软件系统的安全性。

关键词:模糊测试；软件安全；测试框架

中图分类号:TD52 文献标识码:A 文章编号:1008-8725(2010)08-0154-03

Research and Design of Software Security Testing Frame-work Based of Fuzz Testing

YUAN Li-juan[1], ZHANG Yu-yu[2]

(1. College of Information Technology, Baoding University, Baoding 071000, China; 2. Organization Department, Baoding University, Baoding 071000, China)

Abstract: Before the software has not yet formed, the testing has been involved in, prior to this security problem is one of the important elements to consider. Security testing should be planned ahead before the physical software development. In order to identify security risks through software testing, software security vulnerabilities test is divided into security testing plan, along with testing, integration testing in three phases, conceptual and action of two regions. Fuzz is testing more efficient in testing, based on fuzzy test which is suitable for testing for security vulnerabilities and combined with practical test proposes software security testing frame-work so as to identify potential security issues in the whole software process and effectively enhance the security of the software systems.

Key words:fuzz testing; software security; testing frame-work

地,增加一个任务(Create Trigger),来完成这些操作,避免系统运行的失败。通过实行这种改进措施,任务(Query CodeInfo)、任务(Add CodeInfo)、任务(Delete CodeInfo)和任务(Modify CodeInfo)对可靠性需求都有肯定的促进作用。

同时,在角色(Recycle Manager)和任务(Restore Deleted CodeInfo)之间,增加 2 个软目标依赖,即可靠性(Dependability)和安全性(Safety)。这样,在对已删除的信息进行恢复操作时不会产生意想不到的错误，提高系统的安全性与可靠性,如图 3 所示。由于采取了适当的改进措施,系统的可靠性需求得到了满足,同时,可用性需求也得到了满足。

4 结语

可靠性是任何软件系统都必须具备的质量属性，但是，长久以来,人们缺少一个系统化的方法来获取软件的可靠性需求。本文的目的正是为了寻找一种适当的方法来获取软件的可靠性需求。此外,系统的可靠性需求涉及到多方面的原因,本文主要讨论了软件系统的可靠性需求。

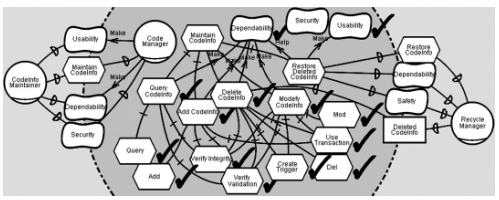

图 3 对策分析

参考文献:

[1] 郑人杰,殷人昆,陶永雷.实用软件工程[M].北京:清华大学出版社,1997.

[2] Mario Barbacci,Mark H. Klein,Thomas A. Longstaff,Charles B. Weinstock. Quality Attributes [EB/OL],Available at http://www.cert.org/archive/pdf/tr021.95.pdf.

[3] Eric S. K. Yu,Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering [EB/OL]. http://www.cs.toronto.edu/pub/eric/RE97.pdf.

[4] Ome3 tool, Available at: http://www.cs.toronto.edu/km/ome/index.htm.

(责任编辑 徐艳杰)

收稿日期:2010-03-22;修订日期:2010-05-19

作者简介:苑立娟(1980-),女,河北保定人,助教,本科,保定学院信息技术系,研究方向:计算机应用技术。

0　前言

随着软件技术的不断发展，从日常生活到各种社会活动，软件已经渗透到了各个领域，软件安全漏洞所造成的危害越来越严重，软件的安全性研究已成为越来越重要的课题，人们开始进行软件测试以达到软件健壮，这其中包括一项容易被测试工作忽略的工作，那就是安全漏洞的测试工作。现今，对于软件的测试的思想多集中在功能性、健壮性等方面。对于软件安全性的测试常常跟软件的通常测试的概念混合在一起，从而，导致软件的测试不全面、不完整，甚至软件测试的严重缺失。如何在现有的软件测试工作中，加入安全性的测试，以避免软件安全漏洞影响整个系统，是一个值得研究的课题[1-2]。

软件安全性问题主要存在于程序书写过程中及运行环境规划中，这使得在软件测试工作中，可以大致划分出 2 个作用域以标示出软件的安全性测试的作用范围。本文针对软件可能的安全性问题进行深入探究，将软件安全作用域划分为局部区域和全局区域。通过静态程序分析和动态测试，结合模糊测试理论，提出基于模糊测试的软件安全性测试架构，作为与普通意义上的一般测试概念并行地测试路线而形成双线测试，从而有效提高软件系统的安全性。

1　模糊测试简介

模糊测试的概念出现有大约二十年的时间了，但是直到最近才引起技术领域的广泛的关注。模糊测试技术的有效应用产生了许多新的工具和日益广泛的影响。

模糊测试属于黑盒技术中的一种可利用的方法，这种方法对于发掘那些用审核方法无法发现的产品关键安全漏洞方面是很有效的。模糊测试的过程是向产品故意地输入无效数据以期望触发错误条件或引起软件或产品的故障。

模糊测试没有一个公认的概念，比较通用的定义是一种通过提供非预期的输入并监视异常结果来发现软件故障的方法。模糊测试是一个自动的或半自动的测试过程，这个过程包括反复向目标软件冲压无序、杂乱、有意识或者无意识的处理数据。模糊器基本上可分为两大类：

⑴ 变异型模糊器：该模糊器通过已有样本形成随机样本，进而形成测试用例；

⑵ 确定型模糊器：该模糊器通过对目标协议或文件格式建模的方法形成测试用例。

模糊测试大体包括以下几个阶段：

⑴ 识别目标。在没有考虑的情况下，要想选择适合的模糊测试工具首先需要清楚目标应用程序。在识别目标应用程序的时候，需要查看以往的测试安全漏洞的历史，这将对识别目标提供很大的帮助，甚至可以确定哪些模块或功能需要进行模糊测试。

⑵ 识别输入。大部分安全漏洞都是因为应用程序在接受用户输入的时候没有排除非法数据或验证其输入的合法性。这种根据样本进行枚举形成的输入向量对模糊测试的成功起着重要的作用。任何形式的输入模型都可以作为，同时也都应该作为测试数据。

⑶ 生成模糊测试数据。形成输入模型之后，模糊测试应该立刻生成。这一过程最好的方法就是自动化方法。

⑷ 执行模糊测试数据。这一阶段通过自动化过程来执行真正的模糊测试。

⑸ 监视异常。该阶段在模糊测试过程中是至关重要的。对故障或异常的监视过程是真正能够获取模糊测试结果的重要手段。即便是向目标程序发送百万级别的模糊数据，有一次导致目标程序崩溃，如果错过了这一次的监视，那么百万级别的模糊测试也是徒劳的。监视的手段可以采用多种形式，并且应该不依赖目标程序和所选择的模糊测试类型。

⑹ 确定可利用性。该阶段需要人工干预，它要求动作执行者具备安全领域的专业知识，以便能够确定获得的模式测试结果能够被利用。

模糊测试的优点在于其成本低、效率高，可自动化执行，且容易配置，是一种有效的测试方法，不光在软件方面，在其它领域同样可以应用模糊测试理论进行相关操作。

模糊测试在测试覆盖率方面仍然处于相对未知的领域，需要进一步的探索和研究。

2　软件安全性的研究与分析

形成软件安全漏洞的元素有很多，各个元素所拥有的加权系数也不相同，需要综合评判。[3]从软件开发的角度大致可以分为：编码阶段形成安全漏洞和执行上下文过程形成安全漏洞。

编码阶段其实是最容易产生安全隐患的阶段，其中包括数学运算漏洞，运算的过程中隐含的一些经常会被忽略的安全漏洞，例如超出设定变量的存放范围、造成溢出等；流程控制漏洞，逻辑控制、逻辑判断等形成的漏洞。边界漏洞例如数组越界。文件格式漏洞，握手程序或模块之间或文件输入、输出过程中形成的漏洞。

执行上下文即软件在运行环境运行时形成安全漏洞。大致可以从以下几个方面考虑：

⑴ 使用者权限漏洞：使用者以不正当的身份进入系统，违法使用权限，造成系统资料的安全与保密隐患，甚至破坏整体系统的正常运行。

⑵ 核心系统监控漏洞，当系统核心的工作环境出现无法配合软件系统需求的状况时，由于不能读取系统资源而造成整个软件系统不能正常运行的安全漏洞。

⑶ 网络安全漏洞；

⑷ 存储设备管理漏洞；

⑸ 资源配置漏洞等。

3　静态测试和动态测试分析

软件的安全隐患可分为局部性的和全局性的。对于局部性安全漏洞的检测，这里主要进行了以下几个方面的分析。

首先，程序员编完代码后需要通过编译器将源码编译为机器码，这一过程中编译器会列出不合语法的错误和部分语意错

误,通过二进制审核软件可以对机器码进行安全漏洞的排查。

其次,程序中逻辑语意和逻辑控制处理过程中也经常会形成安全漏洞。这些安全隐患存在区域最适合用静态测试方法进行漏洞排查,主要过程为 :通过静态测试定义安全漏洞存在区域集 通过静态程序分析器对区域集进行全面分析。

静态测试的着眼点在于逻辑的运行、语意的异常或缺失等方面。静态分析采用直接分析原始程序码的方式,比较适合用于区域性程序安全漏洞的检测工作,一般必须采取与程序语言相关的静态程序分析器,自动化的协助安全漏洞的检测工作[4-5]。

在处理全局性安全漏洞方面,透过静态测试可以找出大部分问题,在软件规格与运行上的安全漏洞称为全局性安全漏洞,此类型安全漏洞不适合利用静态测试,而是必须进行动态测试,才能具体检测出功能需求及运行上的安全漏洞。测试规划与前期准备工作是决定能否找出安全漏洞的关键因素,因此,测试规划需要从异常格式与异常使用、容量与负载、网络传输和硬件资源使用等几个方面进行考虑,根据自己测试目标的实际情况进行取舍后,组合形成动态测试规划。

4 安全软件检测架构

局部性的安全漏洞主要集中在程序单元上,这是安全漏洞检测的前期阶段,该阶段所检测出来的漏洞、漏洞所涉及的范围和区域及其错误报告是进行后续测试的重要参考依据。因此需要进行严格检测工作,以避免安全缺失继续递延至后续的步骤与维护阶段。

因此,最适当的局部性安全漏洞检测应该贯穿于单元测试及集成测试结合。根据软件开发的过程测试可化分为:单元测试、集成测试、系统测试、域测试。安全测试应该贯穿全部测试过程。模糊测试的方法基本上可分为 :基于变异的模糊器(mutation- based fuzzer)对已有数据样本应用变异技术来创建测试用例,基于生成的模糊器(generation- based fuzzer)通过对目标协议或文件格式建模来从头创建测试用例。可将上述 2 种模糊器继续分为预先生成测试用例,随机方法、协议变异人工测试、变异或强制性测试和自动协议生成测试。

预生成测试用例法首先需要对需求的研究,以便理解所有被支持的数据结构和每种数据结构可接受的值范围。随后生成数据包或文件随后对测试边界条件进行测试,或者强制使违法事件发生。这样可以提供测试的精确程度。

随机方法需要生成大量的随机数据然后对目标程序进行攻击。适合用于快速的测试目标系统或目标软件整体性能是否符合需求。

协议变异人工测试需要人工干预而非自动化,这要求测试人员对系统或软件有足够的了解,然后发送违规数据包对目标进行测试,从而获得软件是否存在缺陷或漏洞。

变异或强制性测试是指将有效的协议或数据按照乱序分割数据包或文件中的每一个字节、字、双字或字符串。该方法不需要对目标软件进行彻底研究。该过程可实现自动化。但是要求已经做过测试的那些良好数据包或文件。

自动协议生成测试是一种更高级的强制性测试方法。这种方法需要首先理解和解释协议规约或文件定义。然后创建一个描述协议规范如何工作的文法,这需要识别数据包或文件中的静态部分和动态部分,动态部分是可被模糊的变量。

整个测试流程辅助以安全性测试,在各个测试阶段,5 种模糊测试方法组合搭配,以便每一阶段都能获取漏洞,进而进行弥补,形成的资料和文档也是极其重要的,在最终的验收前的综合测试体系中将是各种测试的依据和参考。详细的结构如图 1 所示。

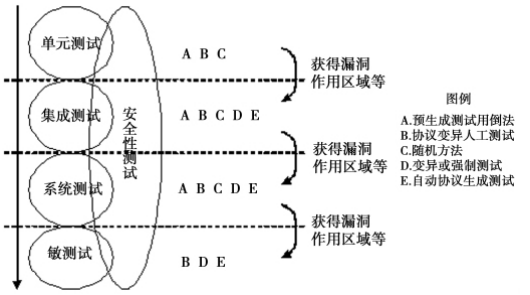

图 1 模糊测试应用架构

全局性漏洞检测需要对整个系统或软件结合运行环境综合考虑,图 1 描述了如何尽可能的在每个测试阶段检测软件安全性的缺陷。全局性漏洞检测是在更高的一个层次上而言,或者是在对图 1 进行适当抽象而得到的一个测试应用框架。每一个阶段获取的漏洞、漏洞的作用区域都代表了该阶段测试的程度,可以通过相关的方法在这个程度上下一个定义,给出一个度量[6],以便当进行回归测试时,通过计算可以给出详细的测试规范[7]。这样对于安全性测试提供了一个指导方向,在某种程度上讲,这个区域是高危的,模糊测试要在这个区域进行多次攻击,这样能够提高回归测试的效率和准确度,也能提升整个系统的安全性测试的效率。

5 结语

软件安全性问题对于软件本身和应用了软件的政府、商业部门等机构的危害越来越严重。网络入侵与系统本身的安全漏洞对当今建立在互联网基础之上的各种模型的危害不断升级,使得各企事业单位的软环境安全性受到前所未有的挑战。基于模糊测试的软件安全性测试架构能够从一定程度上对软件的漏洞进行侦测,进而有效地提高软件的安全品质。其中存在一些需要解决的问题,例如复杂逻辑控制程序单元难以应用模糊测试,则需要通过其它测试手段进行漏洞检测,如何将没有覆盖模糊测试的领域进行有效的漏洞检测是一下步需要研究的课题。

参考文献:

[1] 邓承志,朱卫东.浅谈代码安全质量保障中的模糊测试技术[J].信息网络安全,2009,2:65- 66.
[2] 陈璇.浅谈关于软件安全性测试方法研究[J].电脑知识与技术,2009,5(9):2148- 2149.
[3] 李辉,蔡敏.模糊综合评判在军事运筹中的应用[J].中国新通信,2009,9:76- 77.
[4] 马胜男,孙翊,陈玉忠.软件测试与 Web 服务测试研究进展[J].标准科学,2009,9:80- 83.
[5] 马海云.软件可靠性测试中不确定性问题的研究[J].自动化与仪器仪表,2009,5:128- 129.
[6] 涂玲,周彦晖,张为群.基于模糊推理的软件测试度量方法[J].计算机科学学,2009,36(7):141- 144.
[7] 南光浩. 基于模糊 k 近邻的样本预选取的支持向量机分类算法[J].延边大学学报,2009,35(3):263- 265.

(责任编辑 徐艳杰)