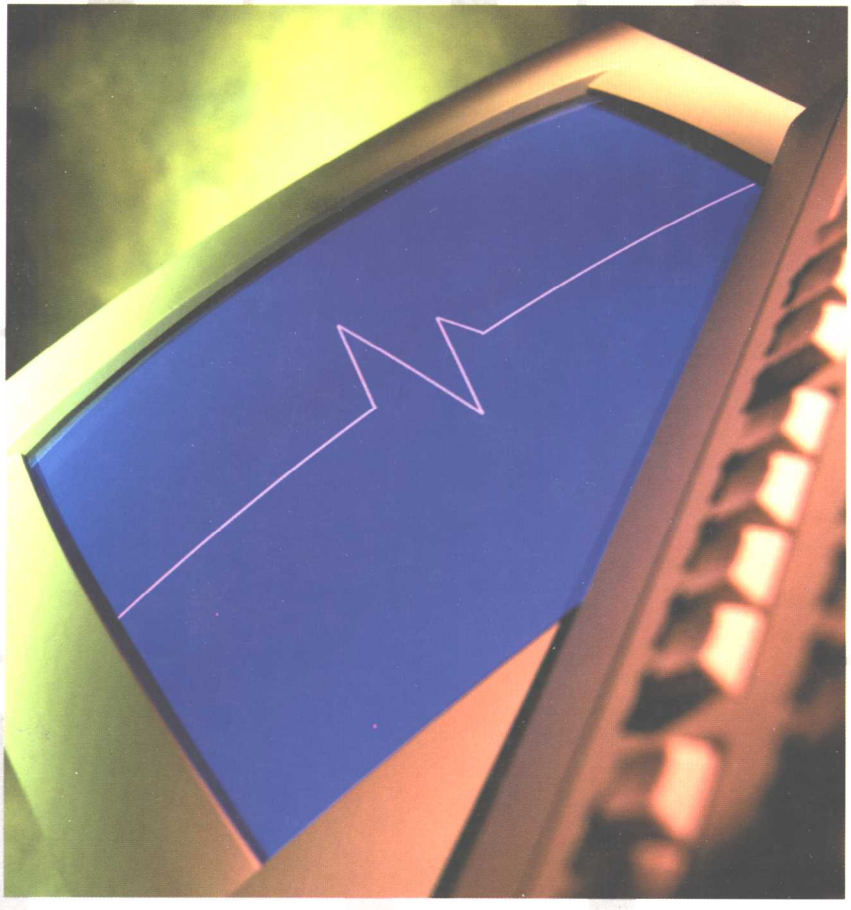
How to Break Software A Practical Guide to Testing

国外IT精品丛书



实用软件测试指南

一本给Microsoft、IBM、Rational等国际公司软件测试部全体人员讲演用的软件测试指导书

〔美〕 James A. Whittaker 著

马良荔 俞立军 译

贲可荣 审校

Addison Wesley

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

国外IT精品丛书

How to Break Software
A Practical Guide to Testing

实用软件测试指南

〔美〕 James A. Whittaker 著

马良荔 俞立军 译

贲可荣 审校

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 提 要

本书所给出的测试并非传统意义上基于书面测试计划实施的循规蹈矩的测试，也没有讨论艰深的测试理论，而是直接面向实际应用，使测试员进行“自由”的测试，提出了对软件进行攻击的思想，从软件的用户界面、文件系统接口和操作系统接口三个最易于攻击的方面来实施攻击，并利用了所开发的软件，帮助测试员简单地捕获异常并强制执行一般错误，最终能更快、更多地发现软件中的错误，改进软件，提高软件质量。

本书可作为计算机专业高年级本科生、计算机专业研究生的软件测试教材或参考书，也可作为软件开发人员、软件测试人员和软件管理人员的参考手册。

版权贸易合同登记号：01-2002-5471

图书在版编目（CIP）数据

实用软件测试指南/（美）维他科（Whittaker, J.A.）著；马良荔等译. - 北京：电子工业出版社，2003.1

书名原文：How to Break Software A Practical Guide to Testing

ISBN 7-5053-8160-1

Ⅰ. 实… Ⅱ. ①维… ②马… Ⅲ. 软件 - 测试 - 指南 Ⅳ. TP311.5-62

中国版本图书馆CIP数据核字（2002）第087529号

责任编辑：吴源　吴月

印　　刷：北京天竺颖华印刷厂

出版发行：电子工业出版社　http://www.phei.com.cn

北京市海淀区万寿路173信箱　邮编：100036

北京市海淀区翠微东里甲2号　邮编：100036

经　　销：各地新华书店

开　　本：787×1092　1/16　印张：13.375　字数：205千字

版　　次：2003年1月第1版　2003年1月第1次印刷

定　　价：22.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换，若书店售缺，请与本社发行部联系。联系电话：（010）68279077

献　辞

与本书包含的技巧相关的测试员和开发人员不胜枚举。但是，有一些人花了好几个小时的时间来讨论、考虑、论述这份材料，我向他们表示我个人的最诚挚的谢意。最感谢微软的James Tierney，Dave Ladd，George Stathakopoulis，Harry Robinson，Bj Rollison，Noel Nyman和Chris Walker；IBM的Mike Houghtaling和Steve Atkin；Rational的Sam Guckenheimer，Andres De Vivanco；Florida技术学院的Ibrahim El-Far，Alan Jorgensen博士，Scott Chase，Florence Mottay，Nikhil Nilakantan，Shirley Becker博士和Cem Kaner博士。

我把这本书主要献给我在Florida技术学院的学生。本书中的这些技术都由这些年轻的（以及不怎么年轻的）、热情的头脑进行过现场测试。我感谢我的这些学生，感谢他们的输入、他们的隐错、他们的热情、他们的挑战、他们的洞察力，他们中的大多数为Florida技术学院成为软件测试圣地立下了汗马功劳。

译 者 序

随着软件技术的不断发展，对软件质量的要求也逐渐提高，如何提高软件可靠性、改进软件过程进而提高软件质量，已经成为国内外许多专家、学者从事的重要研究方向，也成为软件开发机构和运用软件的各类企业迫切关心的问题，而软件测试作为保证软件质量的手段，同时受到了越来越多的重视。

软件测试是一项非常严格的工作，如果能有效地实施软件测试，及时发现软件缺陷和错误，就能有效提高软件质量，改进软件过程。

本书的作者James A. Whittaker一直致力于软件测试方面的研究和实际应用工作，他在本书中所给出的测试并非传统意义上基于书面测试计划实施的循规蹈矩的测试，也没有讨论艰深的测试理论，而是直接面向实际应用，使测试员进行“自由”的测试，提出了对软件进行攻击的思想，从软件的用户界面、文件系统接口和操作系统接口三个最易于攻击的方面来实施攻击，并利用了所开发的软件——运行期间故障植入工具HEAT，即不利环境应用程序测试器（Hostile Environment Application Tester），帮助测试员简单地捕获异常并强制执行一般错误，最终能更快、更多地发现软件中的错误，改进软件，提高软件质量。

我们翻译本书，旨在将测试的新思想奉献给广大读者，希望对广大读者能有所帮助。

本书第1章至第4章由马良荔翻译，其余章节由俞立军翻译。两位译者交叉校阅了对方的部分译文。贲可荣组织了本书的翻译，并对全书进行了审校。

由于各种原因，译稿难免存在错误和疏漏，欢迎读者批评指正。

本书可作为计算机专业高年级本科生、计算机专业研究生的软件测试教材或参考书，亦可作为软件开发人员、软件测试人员和软件管理人员的参考手册。

前　言

关于本书

本书基本上没有包含测试理论。阐述测试理论的书很多，但讲述一个好的测试员如何开展实际测试的书基本上没有。所以我写了这样一本书。我试图进入我所遇到过的最佳测试员的大脑并把最好的技术形成文字，指出隐错在什么地方以及如何最有效地找到这些隐错的方式继而获得可发布的高质量产品。我写本书的第二个目的是使它阅读起来充满趣味性并且包含的技术应用起来也很有趣。我爱中断软件。我希望我的热情将为软件测试行业增添光彩的一页。

关于本书读者

听起来有点像推销，但每一个读者都可以从本书中获得一些他所需要的知识。它以一种吸引初学者的指导风格书写，而最老练的测试人员可能欣赏本书独特的内容。当然，目前我作为大学教授也意味着我在写作本书时要把学生放在心上。我把本书用于本科生和研究生测试课程。学会如何成为一个优秀的测试员并超越基础理论是很重要的。

关于本书的例子

本书给出了来自实际应用软件的真实错误消息和奇异行为（我敢说是“隐错”吗？）的屏幕显示。包含这些例子用来描述本书所讲述的攻击技术。其中有些是大隐错，而有些只是小麻烦。例子的主要目的是教学和说明。这并不意味着仅仅作为大隐错的例子（尽管其中的一些确实如此），也不意味着我们取笑生产

该应用程序的公司。实际上，我精心选择了市场领头羊的应用程序，所以能够被绝大多数读者立即认可。更进一步，每个例子都来自我平常交往的、其开发实践和文化是我所推崇的公司生产的应用程序。

我提醒读者在应用本书中的技术或再现屏幕显示的隐错时要谨慎。许多攻击会使应用程序崩溃。确信在尝试攻击之前保存了你的工作，确保不丢失任何重要的数据。

关于本书的组织

本书由四部分组成。

第一部分是对软件中断研究领域的介绍，以适应讲授对软件和它所工作的复杂环境的理解。对软件测试人员而言，理解软件环境很重要，这样他们就注意到软件内部在做什么以及它是如何与环境交互的。没有这样的理解，很难成为一个成功的软件中断者。

第二部分详细介绍了能通过用户接口应用的特定攻击。无论接口是GUI或是程序的API，这一部分将会给出如何从接口攻击软件，何时应用每个攻击，攻击在破坏软件时为什么会成功以及是怎样成功的。这一部分可能成为在对你的软件目标进行攻击行动时最常引用的部分。

第三部分讨论了非人类系统接口的测试问题（即如何测试应用程序的文件系统、操作系统、软件接口）。讨论了攻击策略。介绍了一个新的称为HEAT的工具，即不利环境应用程序测试器（Hostile Environment Application Tester）。你可以在本书选配光碟中找到“Canned HEAT”。Canned HEAT允许测试员以一种称为运行期故障植入的技术来测试系统接口。

第四部分是结论，结论是软件测试永远不能真正地掌握。相反，它要求不断地学习并获取新的技能。这部分我们描述了在Florida技术学院发生的两件有趣的充满教训的事。你可以把它们用在你的实验室里。它们保证能把学习和娱乐共同融入到任何软件测试方式之中！

附录B～附录D为软件中断者提供了特定的资源。附录B讨论了运行期故障植入。附录C描述了Canned HEAT的使用，Canned HEAT是选配光碟上的一个易

于使用的故障植入工具。附录D重印了在“IEEE Software”上已发表过的一篇文章，它描述了一般的软件测试过程。这些附录补充了本书的内容，可以单独阅读或作为本书的附加材料阅读。附录A包含了常用编程术语的定义。

关于本书的内容

本书脱离了传统的测试。在传统测试中，测试员准备好书面测试规划并把它作为测试软件时的脚本，测试提前规划好并以一种机械的方式执行。本书中的测试技术是灵活的，而传统测试是严格的。如果在一个软件项目中，需求改变了，隐错变成特性，进度压力强制计划重新评估时，就需要灵活性。

软件测试并不是可以预先确定测试什么、执行计划并按计划来实施的一门精确的科学，这需要上帝般的远见。不是靠计划，而是靠智能、洞察力、经验以及对隐错隐藏位置的敏锐判断等来指导测试人员。本书会帮助测试员开发这种洞察力。

作为测试规划和自动测试执行的一个长期拥护者，我对这种自由形式测试的第一个感觉是怀疑。但是，事实简单地驳倒了我。聪明的人做的探索测试找到了所有我见过的最佳隐错。同样是这些聪明的人在最佳测试自动化上也做得更好，不是一个小的零头而是一个数量级。当项目需做困难的、认真的测试时，往往是调入最聪明的测试员而不是测试计划或测试自动化。当需要做出关键的运载决定时，聪明的管理员会忽略测试计划的部署而注重最有经验的测试员的意见。

我不久就确信我那宏大的全自动测试研究议程需要认真地重新考虑。我对那些知道产品并找出了所有隐错的人表示出强烈的兴趣。我可以证实这些民间方法没有脚本并设法抛开了压在桌上文档下的正式的测试计划。被提问时，他们总是回答：“那份计划并未瞄准隐错所在之处。”

本书中体现的技术不仅允许测试员脱离脚本，也鼓励他们这么做。不要盲目相信文档，它们可能是过期的或在产品可测试之前写的。相反，应该开动脑筋！睁大眼睛！想一点，测试一点，再想一点。

但是，不要认为我反对计划或文档。本书简单地讲授测试中的紧张工作计划。不要认为测试自动化是令人沮丧的。有许多重复复杂的任务需要好的工具（确

实，收录在本书选配光碟上的就是这样的工具）。但是，工具永远不能代替智能使用。测试员思考并使用工具收集数据，以帮助他们更加快速有效地探究应用程序。

如果给本书一句题词，则可以是：

熟悉你的产品，考虑你的步调，并让经验引导你。

甚至可以更简单：

开动脑筋，睁开眼睛……测试！

我长期热衷于研究这样的测试。我通过观察人们以及研究他们的隐错来做这项工作。为找出隐错，最好的测试员在做些什么？最佳隐错有什么共同之处？有没有方法推广有经验的测试员的行动使得缺乏经验的测试员能做得更好？

我的意图是捕捉我所遇到的最优秀的测试员的最好思想，并把这些思想形成文字，以这种方式使测试新手能够学到有用的知识并变得更好。如果本书内容未能达到这个目标，那只是因为我个人的沟通技巧不足所致，而不是我的研究主体本身的技巧的反映。本书中好的内容都出自同事们细致研究过的工作。我感谢他们公开了他们找出的隐错以及他们的洞察力。本书中的错误全都归咎于我。把测试的辉煌景象形成文字对于像我这样的凡人来说是很困难的事。

关于本书的补充材料

本书提供了一张选配光碟，它包含了两个非常有用的工具，它们是Florida技术学院的教员和学生编写的。**Canned HEAT**和**Holodeck Lite**是运行期监视和故障植入工具，运行于**Windows NT**系列的平台上。两个工具都很容易使用，在本书的第三部分和附录中有详细的解释。

另外，**www.HowToBreakSoftware.com**是获得最近的工具更新、隐错故事和与破坏软件原理相关的技术公告等信息的在线通道。

James A.Whittaker
Melbourne, Florida
August 2001

各章概要

第一部分：引言

第1章——指导软件测试的故障模型

这一章教你如何考虑软件行为。它给出了有关软件行为的一个新颖观点：通过理解软件在做什么，我们就能够理解如何使它做错。这个认识叫做故障模型，它能引导我们找出隐错的特定攻击，并帮助我们运行那些为了产品成功发布必须做的工作的重要行为。

第二部分：用户接口攻击

第2章——用户接口的测试：输入和输出

这一章可能会是你读得最多并最受益的一章。当你坐在键盘前试图弄清楚应用哪些输入时，本章给出了特定的建议。也告诉你如何使你的软件生成有趣的输出。这一章最适合黑盒测试。

第3章——用户接口的测试：数据和计算

这一章包含了常常被称为“灰盒测试”的内容，因为它处理介于白盒和黑盒之间的问题。尽管不需要源代码，但给出的攻击也告诉了测试员应超越软件接口来理解内部数据和运算。具体攻击的焦点是破坏数据和计算约束。

第三部分：系统接口攻击

第4章——文件系统接口的测试

与数据文件交互时软件会失败。这一章集中于测试文件系统的问题和解决方

法。所附选配光碟提供的工具Canned HEAT在这一章里第一次得到使用。

第5章——软件/操作系统接口测试

软件往往使用其他软件来为它做一些工作。当外部软件资源失败或行为异常时，我们自己的软件也会被破坏。这一章讨论了与应用程序互操作性相关的测试问题，并用Canned HEAT来提供对软件到软件接口测试的洞察力。

第四部分：结论

第6章——临别忠告

有时候并不是技巧或技术造就了了不起的测试小组。这一章讨论了我遇到过的最佳测试小组营造的一种训练方式。他们提出挑战并以富有成效的方式使你的工作更有趣，使你的小组更多产。

附录

附录A——编程术语注释表

曾听开发人员说过并对他们讲的饲草架感到奇怪吗？这一章通过解释诸如API、分隔符、异常处理程序、控件以及谕示等术语消除了神秘感。你可以先阅读这一部分，也可以在本书中出现了不熟悉的术语时进行查阅。

附录B——使用运行期故障植入的测试异常和错误实例

这部分内容有助于读者获得使用Canned HEAT对软件模拟有故障的操作环境时的附加洞察力。尽管本章中的许多建议已经分散在第3章和第4章中，在这里收录是为了有一个全面的参考点。

附录C——使用HEAT：不利环境应用程序测试器

HEAT代表不利环境应用程序测试器（Hostile Environment Application Tester）。它提供了一种机制来简单地跟踪异常（这些术语在附录A中有描述），

并强制执行错误代码。HEAT可以模拟有故障的网络、低内存、装满了的硬盘驱动器或一些其他的失效场景。该附录提供了一个良好格式化的包含在Canned HEAT中的“Help”菜单下的“Help”文件的打印版本。

附录D——什么是软件测试，它为什么这么困难

这是在“IEEE Software”（Vol.17，No.1，Jan/Feb 2000）中发表的一篇文章。收录它是因为它引入本书中没有包括的有关测试的其他阶段（例如，领域分析、基于模型的测试、回归测试以及软件可靠性评估等）。这篇文章试图完整地概述测试软件时遇到的问题并对测试员可获得的解决方法进行总结。带着希望，它会激励你对超越本书范围的有关软件测试的更多知识感兴趣。

致　谢

我要对下列评审人员表示感谢：
Jim Bieman——科罗拉多州立大学
Neil Bitzenhofer——明尼苏达大学
John Callahan——环球软件公司
Barbara Endicott-Popovsky——Endicott咨询有限公司
Bill Junk——爱达荷大学
Gail Kaiser——哥伦比亚大学
Danny Kopec——布鲁克林学院
Bruce Maxim——密歇根大学迪尔伯恩分院
James McDonald——蒙默思郡大学
Rob Sabourin——AmiBug.Com有限公司
Ye Wu——George Mason大学

目 录