

SECURITY SOFTWARE DEVELOPMENT

Assessing and Managing
Security Risks



DOUGLAS A. ASHBAUGH



CRC Press
Taylor & Francis Group

TP309
A819.2

SECURITY SOFTWARE DEVELOPMENT

Assessing and Managing Security Risks



DOUGLAS A. ASHBAUGH, CISSP, CISA



E2009000238



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

AN AUERBACH BOOK

Auerbach Publications
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2009 by Taylor & Francis Group, LLC
Auerbach is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-6380-6 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Ashbaugh, Douglas A.

Security software development : assessing and managing security risks /
Douglas A. Ashbaugh. -- 1st ed.
p. cm.

Includes bibliographical references and index.

ISBN 978-1-4200-6380-6 (alk. paper)

1. Computer security. 2. Application software--Development. 3. Computer networks--Security measures. I. Title.

QA76.9.A25A8246 2008

005.8--dc22

2008015213

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the Auerbach Web site at
<http://www.auerbach-publications.com>

**SECURITY
SOFTWARE
DEVELOPMENT**
Assessing and Managing
Security Risks

Preface

Application security is a relatively new, yet very exciting field. It is being driven by a number of open source, government, regulatory, and industry organizations, but the need for application security is, sadly enough, the fact that software continues to be developed that isn't secure. For example, buffer overflows continue to plague software development despite the fact that buffer overflows and the methods for preventing them have been known for more than 20 years. The author believes that the primary reasons that secure software hasn't been developed lies with two factors:

- First, software development teams have not been sufficiently trained in how to identify vulnerabilities associated with their software development projects.
- Second, software development teams falsely believe that if perimeter security controls are in place, then the software they develop will also be secure, or at least will not affect the perimeter security.

The author was one of those developers who believed that as long as perimeter security (i.e., firewalls, intrusion detection and prevention, anti-virus, etc.) was in place, flaws in his code could not possibly affect the security of that perimeter. That may have been the case when applications were primarily mainframe- or client/server-based. However, the paradigm shifted with the introduction of Web-based applications, as the author painfully discovered.

Traditional firewalls must let Web-based traffic through the perimeter in order for Web-based applications to function. Therefore any attacker who can exploit flaws in the code of a Web application is already within the perimeter! There are additional controls that may be added to secure this perimeter including application and database firewalls, but many organizations have not yet recognized the need for such controls, as headlines sadly continue to point out. When you couple this with the fact that organizations are often slow to adopt new security controls because security is often seen as another expense, it becomes even more imperative for software development teams to understand the vulnerabilities associated with their software development efforts.

The author believes that education is truly the key. Software development teams, including project managers, technical analysts, business analysts, business managers, developers, quality assurance analysts, and testers must all be aware of the vulnerabilities that could plague any software development effort. However, with more than 3400 new vulnerabilities discovered in the first half of 2007 alone, this becomes an almost impossible task.

That is why the author believes in the process of assessing risks within the software development process. Through techniques such as threat modeling, software development teams can quickly begin to learn how to measure the risk associated with their software development projects. Once potential risks are understood management can at least make informed decisions on how to deal with those risks.

It is the sincere hope of the author that you can improve the security of the applications that you develop by following the techniques outlined in this book.

Acknowledgments

The author would like to thank R. J. Droll and Sue Horsman for their assistance in pulling all of the desperate ramblings of the author together to meet the submission deadlines for this manuscript. There is no way I could have done it without your assistance. You're the best!

The author also wishes to thank Jim Bridges, president of Software Engineering Services for taking a chance on a fellow Air Force veteran. You provided me with the chance to prove myself and my abilities at a time in my life when few others would. I truly appreciate all of the faith that you have placed in me and my abilities to get the job done right for you and SES. Thanks Jim!

The author would also like to thank his step-children, Stephanie Bennett and Brody Bennett, for putting up with his surly attitude and closed doors when attempting to put some serious thought into this volume!

Finally, the author would like to thank his wonderful wife Debi for all of the love, support, chiding, and comfort provided during the writing of this book. My love, I could not have done this without your love, patience, understanding—and most of all prodding—to finish writing the book.

And the author also wishes to thank almighty God for providing him with the wisdom and knowledge required to produce this work.

Author Biography

Douglas A. Ashbaugh is a Certified Information Systems Security Professional (CISSP) and member of the International Information Systems Security Certification Consortium (ISC²), as well as a Certified Information Systems Auditor (CISA) and member of the Information Systems Audit and Control Association (ISACA). Mr. Ashbaugh is also the manager of information assurance for Software Engineering Services (SES) where he leads a team of dedicated information security analysts in providing security strategy and solutions, evaluation and assessment services, application security services, and security remediation services to corporate as well as various federal, state, and municipal government clients.

A dedicated information security professional, Mr. Ashbaugh has extensive experience in project management, application development, and information security. His 18+ years of information systems experience in both government and commercial environments provides a solid foundation to achieve outstanding results. He has a Bachelor of Science in engineering operations from Iowa State University. He served eight years in the U.S. Air Force as an acquisition project officer performing project management duties on a number of different development projects ranging in size from \$50,000 to \$3 billion. He has also worked as a software developer/analyst for the financial services industry for a period of more than six years. For the past five years, Mr. Ashbaugh has been providing information security services to a number of clients for SES. SES provides leading-edge IT solutions to DoD, government, state agencies, and the private sector. Mr. Ashbaugh may be reached through the Iowa branch office of Software Engineering Services.

Mr. Ashbaugh is married to a wonderful woman named Debi and lives in the great Midwest with her and a menagerie consisting of two yellow Labrador retrievers, Sam and Barbara Jean, a lop-eared rabbit named Fast-Girl, a red-eared slider (turtle) named Moses, and the head of the household, a black and white tabby named Sassy Marie.

Contents

Preface	xiii
Acknowledgments	xv
Author Biography	xvii
1 Current Trends in Application Security	1
1.1 Recent Data Security Breaches	1
1.2 Definition.....	3
1.3 Legislative and Regulatory Requirements Affecting Application Security	4
1.4 Industry Standards Requiring or Affecting Application Security.....	6
1.5 Risks Associated with Current Trends	10
1.6 Introduction to Test Case That Relates to Current Trends.....	14
1.7 Conclusion	18
References.....	18
2 Security Risk Assessment Methodologies	19
2.1 Definitions.....	20
2.2 Quantitative Risk Assessment Methodologies.....	21
2.2.1 Exposure Factor.....	21
2.2.2 Single Loss Expectancy.....	21
2.2.3 Annualized Rate of Occurrence.....	22
2.2.4 Annualized Loss Expectancy.....	23
2.2.5 Cost-Benefit Analysis.....	23
2.3 Qualitative Risk Assessment Methodologies.....	25
2.3.1 Likelihood of Occurrence.....	26
2.3.2 Magnitude of Impact.....	27
2.3.3 Risk Level.....	28
2.4 Published Methodologies.....	31
2.4.1 Software Engineering Institute's OCTAVE.....	31
2.4.2 STRIDE.....	31
2.4.3 DREAD.....	32
2.4.4 TRIKE.....	33

2.4.5	Australian/New Zealand Standard 4360:2004.....	34
2.4.6	Common Vulnerability Scoring System (CVSS).....	34
2.5	Automated Risk Assessment Tools.....	34
2.6	Tips in Selecting a Methodology	35
2.7	Selecting a Methodology for the Test Case	37
2.7.1	Arguments for Using a Quantitative Risk Analysis Method in the Test Case	38
2.7.2	Arguments against Using a Quantitative Risk Analysis Method in the Test Case	38
2.7.3	Arguments for Using a Qualitative Risk Analysis Method in the Test Case	39
2.7.4	Arguments against Using a Qualitative Risk Analysis Method in the Test Case	39
2.8	Checklist for Deciding on a Security Risk Assessment Methodology	39
2.9	Conclusions	40
3	Identifying Assets.....	41
3.1	Definition.....	42
3.2	Types of Assets Typically Found in Software Development	43
3.2.1	Information Assets	44
3.2.2	External Databases	44
3.2.3	Business Rules	46
3.2.4	Services and Functions	46
3.2.5	Software	46
3.2.6	Proprietary Formulas.....	47
3.2.7	Encryption Software and Encryption Keys	48
3.2.8	People.....	48
3.2.9	Accounts, Transactions, and Calculations	49
3.3	How to Identify Assets in Application Development	49
3.3.1	Business and User Management Involvement.....	49
3.3.2	Review of Organizational Documentation	50
3.3.3	Other Methods of Identifying Assets.....	50
3.4	Determining Assets for the Test Case	52
3.5	Asset Checklist	55
3.6	Summary.....	56
4	Identifying Security Threats	59
4.1	Definition.....	60
4.2	Information Security Threats to Software Development.....	61
4.2.1	Business Threats	61
4.2.2	System Threats.....	62
4.2.3	Human Threats	63
4.2.4	Technical Threats	66

4.2.5	Environmental Threats	70
4.2.6	Natural Threats	72
4.3	How to Identify Security Threats	73
4.3.1	Attack Histories	73
4.3.2	Current Headlines	73
4.3.3	Internet Sites	74
4.3.4	Threat Modeling	75
4.4	Test Case Threats	77
4.4.1	Test Case Business Objectives	78
4.4.2	Test Case User Roles	78
4.4.3	Test Case Use Cases	79
4.4.4	Test Case Components	96
4.4.5	Test Case Architecture	97
4.4.6	Test Case Threats	101
4.5	Conclusion	104
4.6	Threat Identification Checklists	104
4.6.1	Typical Threats (the “Usual Suspects”)	104
4.6.2	Sources of Threat Identification	106
4.6.3	Threat Modeling	106
5	Identifying Vulnerabilities	109
5.1	Definition	109
5.2	The Importance of Identifying Vulnerabilities	110
5.3	Identifying Vulnerabilities	111
5.4	Common Vulnerabilities	113
5.4.1	Buffer Overflows	113
5.4.2	Injection Flaws	113
5.4.3	Information Leakage and Improper Error Handling	115
5.4.4	Cross-Site Scripting	116
5.4.5	Nontechnical Vulnerabilities	117
5.5	Methods of Detecting Vulnerabilities during Software Development	118
5.5.1	Review of Current Controls	119
5.5.2	Code Reviews	119
5.5.3	Testing	120
5.5.4	Static Code Scanning	120
5.5.5	Dynamic Code Scanning	121
5.5.6	Web Application Scanning	121
5.5.7	Network Vulnerability Scanning	121
5.5.8	Review of Best Practice Standards	122
5.6	Secure Coding Techniques to Avoid Vulnerabilities	135
5.6.1	Validate Input	135
5.6.2	Validate Output to Be Displayed on Browsers	135

5.6.3	Keep It Simple.....	136
5.6.4	Follow the Principle of Least Privilege.....	136
5.6.5	Practice Defense in Depth.....	136
5.6.6	Practice Quality Assurance.....	137
5.6.7	Adopt Coding Standards.....	137
5.6.8	Define Security Requirements.....	137
5.6.9	Practice Threat Modeling.....	137
5.7	Vulnerabilities Associated with the Test Case.....	138
5.8	Conclusion.....	140
5.9	Checklists.....	140
5.9.1	Sources of Education about Software Vulnerabilities.....	140
5.9.2	OWASP Top 10 (2007).....	141
5.9.3	SANS Top 20 for 2007.....	141
5.9.4	Methods for Finding Vulnerabilities.....	142
5.9.5	Secure Coding Practices to Avoid Vulnerabilities.....	143
6	Analyzing Security Risks.....	145
6.1	Threat–Vulnerability Pairs.....	146
6.2	Risk Likelihood or Probability.....	147
6.3	Control Analysis.....	152
6.4	Impact or Severity of Threat Actions.....	154
6.4.1	Impact on Confidentiality.....	155
6.4.2	Impact on Integrity.....	155
6.4.3	Impact on Availability.....	156
6.5	Determining Risk Levels.....	158
6.6	Sources of Scales and Tables.....	160
6.7	Determining Security Risks for the Test Case.....	160
6.7.1	Human Threats.....	161
6.7.2	Technical Threats.....	161
6.7.3	Vulnerabilities.....	162
6.7.4	Threat–Action Statements.....	162
6.7.5	Likelihood of Occurrence.....	164
6.7.8	Control Analysis.....	164
6.7.9	Magnitude of Impact.....	166
6.7.10	Risk Levels.....	168
6.8	Conclusion.....	169
6.9	Common Risk Scales and Tables.....	169
6.9.1	Likelihood of Occurrence Scales.....	169
6.9.2	Magnitude of Impact Scales.....	170
6.9.3	Risk Matrixes.....	170
6.9.4	Risk Assessment Reporting Template.....	172
6.9.5	Alternate Risk Assessment Reporting Template.....	175
6.10	Risk Assessment Summary.....	176

6.10.1	Overview	176
6.10.2	OCTAVE Risk Assessment Methodology	177
6.10.3	Identified Assets	177
6.10.4	Critical Assets.....	177
6.10.5	Vulnerability Assessment	178
6.10.6	Security Requirements.....	178
6.10.7	Sources and Potential Impacts of Threats	180
6.10.8	Impact Descriptions	182
6.10.9	Current Protection Strategies.....	184
6.10.10	Risk Analysis.....	186
6.10.11	Risk Mitigation Plans.....	186
6.10.12	Summary.....	187
7	Managing Security Risks	201
7.1	Definitions.....	202
7.2	Risk Mitigation Strategies.....	202
7.2.1	Risk Assumption	203
7.2.2	Risk Transference	203
7.2.3	Risk Avoidance.....	205
7.2.4	Risk Limitation	206
7.3	Protection Strategies	207
7.4	Mitigating Risks in the Test Case.....	209
7.5	Conclusion	211
7.6	Risk Mitigation Checklists	212
7.7	Risk Mitigation Reporting Template.....	213
7.7.1	Risk Mitigation Documentation	213
7.7.2	Risk Mitigation Options	213
7.7.3	Risk Mitigation Strategy	214
7.7.4	Control Implementation Approach.....	215
8	Risk Assessment and Risk Mitigation Activities in the SDLC	217
8.1	Requirements Gathering and Analysis.....	218
8.2	Design	220
8.3	Development	221
8.4	Test.....	222
8.5	Production and Maintenance.....	223
8.6	Risk Management Activities within the Test Case.....	223
8.6.1	Test Case Assets.....	224
8.6.2	Test Case Threats.....	225
8.6.3	Test Case Vulnerabilities	227
8.6.4	Test Case Risks and Mitigation Efforts	228
8.7	Conclusion	230
8.8	Risk Assessment and Risk Mitigation Activity Checklist.....	230

9 Maintaining a Security Risk Assessment and Risk Management

Process	233
9.1 Definitions.....	234
9.2 Risk Management Plans	235
9.3 Supporting Risk Management Practices	238
9.3.1 Top-Down Support	238
9.3.2 Support from Policies and Procedures	240
9.3.3 Legislative, Regulatory, or Compliance Support	241
9.3.4 Certification and Accreditation Support.....	242
9.3.5 Support from Change Management	254
9.4 Continuous Evaluation and Improvement	254
9.4.1 System Security Plan Scope	255
9.4.2 Identifying Key Infrastructure	257
9.4.3 Identification of Key Personnel.....	257
9.4.4 Determining System Boundaries.....	258
9.4.5 Physical Inspections and Walkthroughs	259
9.4.6 Interview Key Personnel.....	259
9.4.7 Incidental Documentation	259
9.4.8 Prepare Documentation	260
9.4.9 Discuss SSP with Management	260
9.4.10 Finalize Documentation.....	261
9.5 Risk Management Policy.....	261
9.6 Conclusions	261
9.7 Risk Management Plan Template	262
9.7.1 Purpose	262
9.7.2 Objective.....	262
9.7.3 References	263
9.7.4 Legal Basis.....	263
9.7.5 Definitions	263
9.7.6 Risk Management Overview	264
9.7.7 Importance of Risk Management	264
9.7.8 Integration of Risk Management into the System Development Life Cycle (SDLC).....	264
9.7.9 Key Roles	264
9.7.10 Risk Assessment	266
9.7.11 Preparing to Assess Risks.....	266
9.7.12 Phase 1: Build Asset-Based Threat Profiles.....	267
9.7.13 Phase 2: Identify Infrastructure Vulnerabilities	267
9.7.14 Phase 3: Develop Security Strategy and Plans	268
9.7.15 Risk Mitigation	268
9.7.16 Risk Mitigation Options	269
9.7.17 Risk Mitigation Strategy	269
9.7.18 Control Implementation Approach.....	270

9.7.19	Evaluation and Assessment.....	271
9.8	Risk Management Policy Template.....	272
9.8.1	Purpose	272
9.8.2	Overview.....	272
9.8.3	Scope.....	273
9.8.4	Statutory Authority	273
9.8.5	Compliance.....	273
9.8.6	Updates	273
9.8.7	Definitions	273
9.8.8	Policy Details: Risk Management.....	274
9.8.9	Integration of Risk Management into the System Development Life Cycle (SDLC).....	274
9.8.10	Key Roles	275
9.8.11	Risk Assessment	276
9.8.12	Risk Mitigation	277
9.8.13	Risk Mitigation Options	278
9.8.14	Risk Mitigation Strategy	278
9.8.15	Control Implementation Approach.....	279
9.8.16	Evaluation and Assessment.....	280
9.9	System Security Plan Template.....	281
9.9.1	Section 1: System Identification.....	281
9.9.2	Section 2: Management Controls	284
9.9.3	Section 3: Operational Controls.....	285
9.9.4	Section 4: Technical Controls.....	288
9.9.5	Section 5: Appendices and Attachments	289
9.9.6	Secure Product Development Policy Template.....	290
Index		299

Chapter 1

Current Trends in Application Security

Information is among the most important assets in any organization. Organizations are constantly building more complex applications to help them accomplish their mission; they are entrusting their sensitive information assets to those applications. But are those information assets secure as they are transmitted, modified, stored, and displayed by those applications? One only has to look at today's headlines to realize that information stored by organizations is not as secure as it could, or should, be.

1.1 Recent Data Security Breaches

Let's look at some of the recent data security breaches in the news today:

- July 27th, 2007: City Harvest, New York. Improper access to systems that contained donor credit card information resulted in the improper exposure of approximately 12,000 records.
- July 26th, 2007: Names and Social Security numbers of 10,554 U.S. Marines were found through the Google Internet search engine.
- July 25th, 2007: The private medical information, including Social Security numbers and treatment details of 25 people who sought medical assistance from the county was posted on the Hidalgo County, Texas Web site.

2 ■ *Security Software Development*

- July 24th, 2007: A security lapse compromised names, addresses, and Social Security numbers of more than 51,000 employees and patients of St. Vincent's Hospital in Indianapolis, Indiana.
- July 23rd, 2007: A security hole on a Fox News Web server exposed sensitive content to the public, including log-in information that allowed hackers to access names, phone numbers, and e-mail addresses of at least 1.5 million people.
- July 21st, 2007: University of Michigan databases were hacked. More than 5500 names, addresses, Social Security numbers, birth dates, and in some cases, the school districts where former students were teaching were exposed.
- July 20th, 2007: A Pentagon contractor may have compromised personal information, such as names, addresses, birth dates, Social Security numbers, and health information about 580,000 military personnel and their relatives because it did not encrypt data transmitted online.

These incidents represent just one week's worth of recent incidents reported by the Privacy Rights Clearinghouse (PRC) (<http://www.privacyrights.org/>) with a grand total of 2,159,079 records that were potentially compromised. The PRC has reported a total of 218,621,856 compromised records since the beginning of 2005. In reality, the number is probably much larger, because for many of the breaches listed by the PRC, the number of records actually compromised is unknown, and there are many data breaches that go unreported.

By conservative estimates at least 230 million records held by private companies, private and public organizations, universities, state and local governments, and the federal government have been compromised over the past three years. How were these records compromised? What security controls have failed to protect this valuable resource called information? The answer is that many different controls have failed. Some breaches are caused by the simple loss or theft of media containing confidential information. Theft (or misplacement) of laptops, hard drives, flash drives, backup tapes, and CD/DVD ROM account for many of the data breaches. Still others are caused by operator error, improperly configured or protected systems, improperly or poorly trained people, and transmission of information in the clear or just plain ignorance. Finally, many of these record losses can be attributed at least indirectly to poor, inconsistent, or nonexistent application security.

What is application security? Inasmuch as this book is about secure software development—which means that it is really all about application security—a definition is in order.