

Table of Contents



Foreword.....	xix
Introduction.....	xxi
Who Is This Book For?.....	xxii
Organization of This Book.....	xxiii
System Requirements.....	xxiii
Technology Updates.....	xxiii
Code Samples and Companion Content.....	xxiv
Support for This Book.....	xxiv
Questions and Comments.....	xxiv
Acknowledgments.....	xxiv
1 General Approach to Security Testing.....	1
Different Types of Security Testers.....	2
An Approach to Security Testing.....	3
Understanding Deeply What You Are Testing.....	4
Thinking Maliciously About Your Target.....	6
Attacking the Product.....	8
Stay Informed About New Attacks.....	8
Summary.....	9
2 Using Threat Models for Security Testing.....	11
Threat Modeling.....	11
How Testers Can Leverage a Threat Model.....	12
Data Flow Diagrams.....	13
Enumeration of Entry Points and Exit Points.....	14
Enumeration of Threats.....	15
How Testers Should Use a Completed Threat Model.....	18
Threats May Be Erroneously Dismissed.....	18
Implementation Rarely Matches the Specification or Threat Model.....	21
Summary.....	22

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

3	Finding Entry Points	23
	Finding and Ranking Entry Points	24
	Assessing the Risk of Entry Points	24
	Common Entry Points	24
	Files	25
	Sockets	29
	HTTP Requests	31
	Named Pipes	34
	Pluggable Protocol Handlers	38
	Malicious Server Responses	39
	Programmatic Interfaces	40
	SQL	41
	Registry	41
	User Interfaces	44
	E-mail	45
	Command-Line Arguments	47
	Environment Variables	48
	Summary	50
4	Becoming a Malicious Client	51
	Client/Server Interaction	51
	Finding Requests the Server Normally Accepts	52
	Manipulating Network Requests	54
	Testing HTTP	58
	Understanding a Stateless Protocol	59
	Testing Methods of Receiving Input	59
	Testing Specific Network Requests Quickly	70
	Testing Tips	71
	Summary	72
5	Becoming a Malicious Server	73
	Understanding Common Ways Clients Receive Malicious Server Responses	74
	Does SSL Prevent Malicious Server Attacks?	76
	Manipulating Server Responses	77
	Common Vulnerabilities Found When Sending Malicious Responses	77

Examples of Malicious Response Bugs	77
Example: Telnet Client Environment Variable Disclosure	78
Example: File Caching Allows Arbitrary Code Execution	79
Myth: It Is Difficult for an Attacker to Create a Malicious Server	80
EvilServer	80
Understanding Downgrade MITM Attacks	80
Testing Tips	81
Summary	82
6 Spoofing	83
Grasping the Importance of Spoofing Issues	83
Caller ID Spoofing	83
Finding Spoofing Issues	85
General Spoofing	85
IP Address Spoofing	86
MAC Address Spoofing	87
Spoofing Using Network Protocols	88
User Interface Spoofing	91
Rewording Dialog Boxes	91
Reformatting Using Control Characters	93
Z-Order Spoofing	96
Misleading URLs and Filenames	97
Testing Tips	100
Summary	101
7 Information Disclosure	103
Problems with Information Disclosure	103
Locating Common Areas of Information Disclosure	104
Disclosure in Files	104
Disclosures over a Network	113
Identifying Interesting Data	117
Obfuscating Data	117
Implied Disclosures	118
Summary	119

8	Buffer Overflows and Stack and Heap Manipulation	121
	Understanding How Overflows Work	124
	Stack Overflows	125
	Integer Overflows	129
	Heap Overruns	136
	Other Attacks	138
	Testing for Overruns: Where to Look for Cases	138
	Network	138
	Documents and Files	139
	Information Shared Between Users with Higher and Lower Privileges	139
	Programmable Interfaces	140
	Black Box (Functional) Testing	141
	Determining What Data Is Expected	141
	Using Data You Recognize	142
	Knowing the Limits and Bounds	142
	Maintaining Overall Data Integrity	144
	Strategies for Transforming Normal Data into Overruns	148
	Testing Both Primary and Secondary Actions	151
	What to Look For	152
	Runtime Tools	163
	Fuzzing	165
	White Box Testing	166
	Things to Look For	167
	Overflow Exploitability	171
	Unicode Data	176
	Filtered Data	176
	Additional Topics	177
	Noncode Execution Overflows Can Be Serious, Too	177
	/GS Compiler Switch	179
	Testing Tips	182
	Summary	183
9	Format String Attacks	185
	What Are Format Strings?	186
	Understanding Why Format Strings Are a Problem	186
	Anatomy of a printf Call	187
	Misinterpreting the Stack	188
	Overwriting Memory	190

Testing for Format String Vulnerabilities.....	191
Reviewing Code	192
Black Box Testing	193
Walkthrough: Seeing a Format String Attack in Action	194
Finding the Format String Bug	194
Analyzing Exploitability	195
Digging Deeper: Working Around Exploitability Problems.....	197
Building a Simple Payload.....	210
Testing Tips	217
Summary	218
10 HTML Scripting Attacks	219
Understanding Reflected Cross-Site Scripting Attacks Against Servers.....	220
Example: Reflected XSS in a Search Engine	220
Understanding Why XSS Attacks Are a Security Concern	223
Exploiting Server-Reflected XSS Bugs	225
POSTs Are Exploitable, Too	226
Understanding Persistent XSS Attacks Against Servers	228
Example: Persistent XSS in a Web Guestbook	228
Exploiting Persistent XSS Against Servers.....	230
Identifying Attackable Data for Reflected and Persistent XSS Attacks.....	230
Common Ways Programmers Try to Stop Attacks	232
HTML-Encoded Data Doesn't Always Stop the Attack	233
Understanding Reflected XSS Attacks Against Local Files	236
Example: Local HTML File Reflected XSS	236
Exploiting Reflected XSS Bugs in Local Files.....	237
Understanding Why Local XSS Bugs Are an Issue.....	238
Using Local XSS Bugs to Run Binaries on the Victim's Machine.....	240
HTML Resources	241
Compiled Help Files.....	243
Finding XSS Bugs in Client-Side Script	244
Understanding Script Injection Attacks in the My Computer Zone	246
Example: Script Injection in Winamp Playlist.....	246
Non-HTML Files Parsed as HTML	248
Ways Programmers Try to Prevent HTML Scripting Attacks	251
Filters	251

	Gaining In-Depth Understanding of the Browser's Parser	253
	Comments in Styles	254
	ASP.NET Built-in Filters	255
	Understanding How Internet Explorer Mitigates XSS	
	Attacks Against Local Files	256
	Links from the Internet to the My Computer Zone Are Blocked	256
	Script Disabled in the My Computer Zone by Default	257
	Identifying HTML Scripting Vulnerabilities	258
	Finding HTML Scripting Bugs Through Code Review	259
	Identifying All Places Content Is Returned to the Web Browser	
	or File System	259
	Determining Whether Output Contains Attacker-Supplied Data	259
	Verifying That Attacker Data Is Properly Validated and/or Encoded	260
	ASP.NET Automatically Encodes the Data...Sometimes	261
	Summary	262
11	XML Issues	263
	Testing Non-XML Security Issues in XML Input Files	263
	Well-Formed XML	264
	Valid XML	264
	Including Nonalphanumeric Data in XML Input	265
	Testing XML-Specific Attacks	268
	Entities	268
	XML Injection	270
	Large File References	273
	Simple Object Access Protocol	273
	Testing SOAP	276
	Testing Tips	277
	Summary	278
12	Canonicalization Issues	279
	Understanding the Importance of Canonicalization Issues	279
	Finding Canonicalization Issues	280
	File-Based Canonicalization Issues	280
	Directory Traversal	281
	Defeating Filename Extension Checks	282
	Other Common Mistakes That Lead to Canonicalization Issues	285

Web-Based Canonicalization Issues	290
Encoding Issues	290
URL Issues	295
Testing Tips	298
Summary	299
13 Finding Weak Permissions	301
Understanding the Importance of Permissions	301
Finding Permissions Problems	303
Understanding the Windows Access Control Mechanism	304
What Is a Securable Object?	304
What Is a Security Descriptor?	305
What Is an ACL?	305
What Is an ACE?	306
Finding and Analyzing Permissions on Objects	307
Using the Windows Security Properties Dialog Box	307
Using AccessEnum	309
Using Process Explorer	309
Using ObjSD	311
Using AppVerifier	312
Recognizing Common Permissions Problems	312
Weak DACLs	312
NULL DACLs	317
Improper Ordering of ACEs	318
Object Creator	318
Accessing Resources Indirectly	319
Forgetting to Revert Permissions	319
Squatting Attacks	320
Exploiting Race Conditions	320
File Links	322
Determining the Accessibility of Objects	325
Remotely Accessible Objects	325
Locally Accessible Objects	327
Other Permissions Considerations	328
.NET Permissions	328
SQL Permissions	328
Role-Based Security	331
Summary	332

14	Denial of Service Attacks	333
	Understanding Types of DoS Attacks	333
	Finding Implementation Flaws	334
	Finding Resource Consumption Flaws	340
	Finding Solutions for a Hard Problem	347
	Testing Tips	348
	Summary	348
15	Managed Code Issues	349
	Dispelling Common Myths About Using Managed Code	350
	Myth 1: Buffer Overflows Don't Exist in Managed Code	350
	Myth 2: ASP.NET Web Controls Prevent Cross-Site Scripting	350
	Myth 3: Garbage Collection Prevents Memory Leaks	351
	Myth 4: Managed Code Prevents SQL Injection	352
	Understanding the Basics of Code Access Security	352
	User Security vs. Code Security	352
	Overview of CAS	353
	Assemblies	354
	Evidence	355
	Permissions	355
	Policies	356
	Global Assembly Cache	360
	Stack Walks	360
	Stack Walk Modifiers	362
	Finding Problems Using Code Reviews	365
	Calling Unsafe Code	366
	Finding Problems with Asserts	368
	Finding Problems with Link Demands	370
	Recognizing Poor Exception Handling	372
	Understanding the Issues of Using APTCA	375
	Decompiling .NET Assemblies	381
	Testing Tips	382
	Summary	383
16	SQL Injection	385
	Exactly What Is SQL Injection?	385
	Understanding the Importance of SQL Injection	387
	Assessing the Vulnerability of Applications	388

Finding SQL Injection Issues	388
Using a Black Box Testing Approach	389
Using Code Reviews	400
Avoiding Common Mistakes About SQL Injection	403
Escape Single Quotation Marks in Input	403
Remove Semicolons to Block Multiple Statements	404
Use Only Stored Procedures	405
Remove Unwanted Stored Procedures	405
Place the Computer That Runs SQL Server Behind a Firewall	406
Understanding Repurposing of SQL Stored Procedures	407
Example: Backing Up Documents	407
Hunting for Stored Procedure Repurposing Issues	408
Recognizing Similar Injection Attacks	409
Testing Tips	409
Summary	410
17 Observation and Reverse Engineering	411
Observation Without a Debugger or Disassembler	411
Comparing Output	412
Using Monitoring Tools	413
Using a Debugger to Trace Program Execution and Change its Behavior	415
Modifying Execution Flow to Bypass Restrictions	415
Reading and Modifying Memory Contents Under a Debugger	420
Using a Decompiler or Disassembler to Reverse Engineer a Program	424
Understanding Differences Between Native Code and Bytecode Binaries	425
Spotting Insecure Function Calls Without Source Code	427
Reverse Engineering Algorithms to Identify Security Flaws	431
Analyzing Security Updates	434
Testing Tips	435
Legal Considerations	436
Summary	436
18 ActiveX Repurposing Attacks	437
Understanding ActiveX Controls	438
Creating ActiveX Controls in Internet Explorer	438
Initializing and Scripting ActiveX Controls	440
Repurposing ActiveX Controls	441
Understanding the ActiveX Control Security Model	445

	Using the ActiveX Control Testing Methodology	451
	Additional Testing Tricks and Techniques	458
	ActiveX Control Testing Walkthrough.	467
	<i>Clear</i>	468
	<i>ClipboardCopy</i>	470
	<i>ClipboardPaste</i>	470
	<i>InvokeRTFEditor</i>	472
	<i>LoadRTF</i>	480
	<i>NumChars</i>	482
	<i>RTFEditor</i> Property.	482
	<i>RTFEditor</i> PARAM.	484
	<i>RTFEditorOverride</i>	484
	Challenge	486
	Testing Tips	486
	Summary	487
19	Additional Repurposing Attacks	489
	Understanding Document Formats That Request External Data	489
	Common Mitigation for Document Formats	
	Requesting External Data	490
	Testing Document Formats That Request External Data	491
	Web Pages Requesting External Data.	492
	CSRF Through Query String URLs	492
	CSRF Through POST Data.	493
	Common Ways to Prevent CSRF Attacks	494
	CSRF Through SOAP Data	495
	Testing for CSRF Attacks	496
	Understanding Repurposing of Window and Thread Messages	496
	Testing for Shatter Attacks	497
	Summary	497
20	Reporting Security Bugs	499
	Reporting the Issue	499
	Contacting the Vendor	500
	What to Expect After Contacting the Vendor	502
	Dealing with Unresponsive Vendors.	502

Public Disclosure	503
Deciding on the Amount of Detail	503
Timing the Disclosure	504
Addressing Security Bugs in Your Product	504
Communicating with Bug Finders	505
Identifying the Root Cause	505
Looking for Related Bugs	505
Determining Affected Products and Versions	506
Testing the Fix	506
Determining Mitigations and Workarounds	506
Releasing Patches Simultaneously for All Affected Products and Versions	506
Summary	507
A Tools of the Trade	509
B Security Test Cases Cheat Sheet	517
Network Requests and Responses	517
Spoofing	518
Information Disclosures	519
Buffer Overflows	520
Format Strings	521
Cross-Site Scripting and Script Injection	521
XML	522
SOAP	523
Canonicalization Issues	523
Weak Permissions	526
Denial of Service	526
Managed Code	527
SQL Injection	528
ActiveX	528
Index	531

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/