

2ND EDITION

HACKING

THE ART OF EXPLOITATION

JON ERICKSON



San Francisco

HACKING: THE ART OF EXPLOITATION, 2ND EDITION. Copyright © 2008 by Jon Erickson.

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.



Printed on recycled paper in the United States of America

11 10 09 08 07 1 2 3 4 5 6 7 8 9

ISBN-10: 1-59327-144-1

ISBN-13: 978-1-59327-144-2

Publisher: William Pollock

Production Editors: Christina Samuell and Megan Dunchak

Cover Design: Octopod Studios

Developmental Editor: Tyler Ortman

Technical Reviewer: Aaron Adams

Copyeditors: Dmitry Kirsanov and Megan Dunchak

Compositors: Christina Samuell and Kathleen Mish

Proofreader: Jim Brook

Indexer: Nancy Guenther

For information on book distributors or translations, please contact No Starch Press, Inc. directly:

No Starch Press, Inc.

555 De Haro Street, Suite 250, San Francisco, CA 94107

phone: 415.863.9900; fax: 415.863.9950; info@nostarch.com; www.nostarch.com

Library of Congress Cataloging-in-Publication Data

Erickson, Jon, 1977-

Hacking : the art of exploitation / Jon Erickson. -- 2nd ed.

p. cm.

ISBN-13: 978-1-59327-144-2

ISBN-10: 1-59327-144-1

1. Computer security. 2. Computer hackers. 3. Computer networks--Security measures. I. Title.

QA76.9.A25E75 2008

005.8--dc22

2007042910

No Starch Press and the No Starch Press logo are registered trademarks of No Starch Press, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of this work, neither the author nor No Starch Press, Inc. shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

BRIEF CONTENTS

Preface	xi
Acknowledgments	xii
0x100 Introduction	1
0x200 Programming	5
0x300 Exploitation	115
0x400 Networking	195
0x500 Shellcode	281
0x600 Countermeasures.....	319
0x700 Cryptology	393
0x800 Conclusion	451
Index	455

This page intentionally left blank

C O N T E N T S I N D E T A I L

PREFACE	xi
ACKNOWLEDGMENTS	xii
0x100 INTRODUCTION	1
0x200 PROGRAMMING	5
0x210 What Is Programming?	6
0x220 Pseudo-code	7
0x230 Control Structures	8
0x231 If-Then-Else.....	8
0x232 While/Until Loops	9
0x233 For Loops	10
0x240 More Fundamental Programming Concepts	11
0x241 Variables	11
0x242 Arithmetic Operators	12
0x243 Comparison Operators	14
0x244 Functions.....	16
0x250 Getting Your Hands Dirty	19
0x251 The Bigger Picture	20
0x252 The x86 Processor	23
0x253 Assembly Language.....	25
0x260 Back to Basics.....	37
0x261 Strings	38
0x262 Signed, Unsigned, Long, and Short	41
0x263 Pointers.....	43
0x264 Format Strings.....	48
0x265 Typecasting	51
0x266 Command-Line Arguments	58
0x267 Variable Scoping	62
0x270 Memory Segmentation	69
0x271 Memory Segments in C	75
0x272 Using the Heap	77
0x273 Error-Checked malloc()	80
0x280 Building on Basics	81
0x281 File Access	81
0x282 File Permissions	87
0x283 User IDs	88
0x284 Structs.....	96
0x285 Function Pointers	100
0x286 Pseudo-random Numbers	101
0x287 A Game of Chance	102

0x300 EXPLOITATION**115**

0x310	Generalized Exploit Techniques	118
0x320	Buffer Overflows	119
	0x321 Stack-Based Buffer Overflow Vulnerabilities	122
0x330	Experimenting with BASH.....	133
	0x331 Using the Environment.....	142
0x340	Overflows in Other Segments	150
	0x341 A Basic Heap-Based Overflow	150
	0x342 Overflowing Function Pointers.....	156
0x350	Format Strings.....	167
	0x351 Format Parameters.....	167
	0x352 The Format String Vulnerability.....	170
	0x353 Reading from Arbitrary Memory Addresses	172
	0x354 Writing to Arbitrary Memory Addresses	173
	0x355 Direct Parameter Access	180
	0x356 Using Short Writes	182
	0x357 Detours with .dtors.....	184
	0x358 Another notesearch Vulnerability	189
	0x359 Overwriting the Global Offset Table	190

0x400 NETWORKING**195**

0x410	OSI Model	196
0x420	Sockets	198
	0x421 Socket Functions.....	199
	0x422 Socket Addresses	200
	0x423 Network Byte Order	202
	0x424 Internet Address Conversion	203
	0x425 A Simple Server Example	203
	0x426 A Web Client Example	207
	0x427 A Tinyweb Server.....	213
0x430	Peeling Back the Lower Layers.....	217
	0x431 Data-Link Layer.....	218
	0x432 Network Layer	220
	0x433 Transport Layer	221
0x440	Network Sniffing	224
	0x441 Raw Socket Sniffer.....	226
	0x442 libpcap Sniffer	228
	0x443 Decoding the Layers	230
	0x444 Active Sniffing.....	239
0x450	Denial of Service	251
	0x451 SYN Flooding	252
	0x452 The Ping of Death.....	256
	0x453 Teardrop	256
	0x454 Ping Flooding	257
	0x455 Amplification Attacks	257
	0x456 Distributed DoS Flooding.....	258
0x460	TCP/IP Hijacking.....	258
	0x461 RST Hijacking	259
	0x462 Continued Hijacking	263

0x470	Port Scanning	264
	0x471 Stealth SYN Scan	264
	0x472 FIN, X-mas, and Null Scans	264
	0x473 Spoofing Decoys	265
	0x474 Idle Scanning.....	265
	0x475 Proactive Defense (shroud).....	267
0x480	Reach Out and Hack Someone	272
	0x481 Analysis with GDB.....	273
	0x482 Almost Only Counts with Hand Grenades	275
	0x483 Port-Binding Shellcode	278

0x500 SHELLCODE

281

0x510	Assembly vs. C	282
	0x511 Linux System Calls in Assembly	284
0x520	The Path to Shellcode.....	286
	0x521 Assembly Instructions Using the Stack	287
	0x522 Investigating with GDB.....	289
	0x523 Removing Null Bytes	290
0x530	Shell-Spawning Shellcode.....	295
	0x531 A Matter of Privilege.....	299
	0x532 And Smaller Still.....	302
0x540	Port-Binding Shellcode	303
	0x541 Duplicating Standard File Descriptors.....	307
	0x542 Branching Control Structures	309
0x550	Connect-Back Shellcode	314

0x600 COUNTERMEASURES

319

0x610	Countermeasures That Detect	320
0x620	System Daemons	321
	0x621 Crash Course in Signals.....	322
	0x622 Tinyweb Daemon	324
0x630	Tools of the Trade.....	328
	0x631 tinywebd Exploit Tool.....	329
0x640	Log Files.....	334
	0x641 Blend In with the Crowd	334
0x650	Overlooking the Obvious	336
	0x651 One Step at a Time	336
	0x652 Putting Things Back Together Again	340
	0x653 Child Laborers	346
0x660	Advanced Camouflage	348
	0x661 Spoofing the Logged IP Address	348
	0x662 Logless Exploitation	352
0x670	The Whole Infrastructure	354
	0x671 Socket Reuse	355
0x680	Payload Smuggling	359
	0x681 String Encoding	359
	0x682 How to Hide a Sled	362
0x690	Buffer Restrictions	363
	0x691 Polymorphic Printable ASCII Shellcode.....	366

0x6a0	Hardening Countermeasures.....	376
0x6b0	Nonexecutable Stack	376
	0x6b1 ret2libc	376
	0x6b2 Returning into system().....	377
0x6c0	Randomized Stack Space	379
	0x6c1 Investigations with BASH and GDB	380
	0x6c2 Bouncing Off linux-gate	384
	0x6c3 Applied Knowledge	388
	0x6c4 A First Attempt.....	388
	0x6c5 Playing the Odds.....	390

0x700 CRYPTOLOGY 393

0x710	Information Theory	394
	0x711 Unconditional Security	394
	0x712 One-Time Pads.....	395
	0x713 Quantum Key Distribution.....	395
	0x714 Computational Security	396
0x720	Algorithmic Run Time	397
	0x721 Asymptotic Notation	398
0x730	Symmetric Encryption.....	398
	0x731 Lov Grover's Quantum Search Algorithm.....	399
0x740	Asymmetric Encryption.....	400
	0x741 RSA	400
	0x742 Peter Shor's Quantum Factoring Algorithm	404
0x750	Hybrid Ciphers	406
	0x751 Man-in-the-Middle Attacks	406
	0x752 Differing SSH Protocol Host Fingerprints	410
	0x753 Fuzzy Fingerprints	413
0x760	Password Cracking.....	418
	0x761 Dictionary Attacks	419
	0x762 Exhaustive Brute-Force Attacks.....	422
	0x763 Hash Lookup Table	423
	0x764 Password Probability Matrix	424
0x770	Wireless 802.11b Encryption	433
	0x771 Wired Equivalent Privacy	434
	0x772 RC4 Stream Cipher	435
0x780	WEP Attacks	436
	0x781 Offline Brute-Force Attacks.....	436
	0x782 Keystream Reuse	437
	0x783 IV-Based Decryption Dictionary Tables	438
	0x784 IP Redirection.....	438
	0x785 Fluhrer, Mantin, and Shamir Attack	439

0x800 CONCLUSION 451

0x810	References.....	452
0x820	Sources	454

INDEX 455