

Computer Viruses and Malware

Advances in Information Security

Sushil Jajodia

Consulting Editor

Center for Secure Information Systems

George Mason University

Fairfax, VA 22030-4444

email: jajodia@gmu.edu

The goals of the Springer International Series on ADVANCES IN INFORMATION SECURITY are, one, to establish the state of the art of, and set the course for future research in information security and, two, to serve as a central reference source for advanced and timely topics in information security research and development. The scope of this series includes all aspects of computer and network security and related areas such as fault tolerance and software assurance.

ADVANCES IN INFORMATION SECURITY aims to publish thorough and cohesive overviews of specific topics in information security, as well as works that are larger in scope or that contain more detailed background information than can be accommodated in shorter survey articles. The series also serves as a forum for topics that may not have reached a level of maturity to warrant a comprehensive textbook treatment.

Researchers, as well as developers, are encouraged to contact Professor Sushil Jajodia with ideas for books under this series.

Additional titles in the series:

HOP INTEGRITY IN THE INTERNET by Chin-Tser Huang and Mohamed G. Gouda; ISBN-10: 0-387-22426-3

PRIVACY PRESERVING DATA MINING by Jaideep Vaidya, Chris Clifton and Michael Zhu; ISBN-10: 0-387-25886-8

BIOMETRIC USER AUTHENTICATION FOR IT SECURITY: From Fundamentals to Handwriting by Claus Vielhauer; ISBN-10: 0-387-26194-X

IMPACTS AND RISK ASSESSMENT OF TECHNOLOGY FOR INTERNET SECURITY: Enabled Information Small-Medium Enterprises (TEISMES) by Charles A. Shoniregun; ISBN-10: 0-387-24343-7

SECURITY IN E-LEARNING by Edgar R. Weippl; ISBN: 0-387-24341-0

IMAGE AND VIDEO ENCRYPTION: From Digital Rights Management to Secured Personal Communication by Andreas Uhl and Andreas Pommer; ISBN: 0-387-23402-0

INTRUSION DETECTION AND CORRELATION: Challenges and Solutions by Christopher Kruegel, Fredrik Valeur and Giovanni Vigna; ISBN: 0-387-23398-9

THE AUSTIN PROTOCOL COMPILER by Tommy M. McGuire and Mohamed G. Gouda; ISBN: 0-387-23227-3

ECONOMICS OF INFORMATION SECURITY by L. Jean Camp and Stephen Lewis; ISBN: 1-4020-8089-1

PRIMALITY TESTING AND INTEGER FACTORIZATION IN PUBLIC KEY CRYPTOGRAPHY by Song Y. Yan; ISBN: 1-4020-7649-5

SYNCHRONIZING E-SECURITY by Godfried B. Williams; ISBN: 1-4020-7646-0

Additional information about this series can be obtained from <http://www.springeronline.com>

Computer Viruses and Malware

by

John Aycock
University of Calgary
Canada

 Springer

John Aycock
University of Calgary
Dept. Computer Science
2500 University Drive N.W.
CALGARY AB T2N 1N4
CANADA

Library of Congress Control Number: 2006925091

Computer Viruses and Malware
by John Aycock, University of Calgary, AB, Canada

ISBN-13: 978-0-387-30236-2
ISBN-10: 0-387-30236-0
e-ISBN-13: 978-0-387-34188-0
e-ISBN-10: 0-387-34188-9

Printed on acid-free paper.

The use of general descriptive names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

© 2006 Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springer.com

*To all the two-legged critters
in my house*

Contents

Dedication	v
List of Figures	xi
Preface	xv
1. WE'VE GOT PROBLEMS	1
1.1 Dramatis Personae	1
1.2 The Myth of Absolute Security	2
1.3 The Cost of Malware	3
1.4 The Number of Threats	4
1.5 Speed of Propagation	5
1.6 People	6
1.7 About this Book	7
1.8 Some Words of Warning	7
2. DEFINITIONS AND TIMELINE	11
2.1 Malware Types	11
2.1.1 Logic Bomb	12
2.1.2 Trojan Horse	12
2.1.3 Back Door	13
2.1.4 Virus	14
2.1.5 Worm	15
2.1.6 Rabbit	16
2.1.7 Spyware	16
2.1.8 Adware	17
2.1.9 Hybrids, Droppers, and Blended Threats	17
2.1.10 Zombies	18
2.2 Naming	19

2.3	Authorship	21
2.4	Timeline	22
3.	VIRUSES	27
3.1	Classification by Target	28
3.1.1	Boot-Sector Infectors	28
3.1.2	File Infectors	30
3.1.3	Macro Viruses	33
3.2	Classification by Concealment Strategy	34
3.2.1	No Concealment	34
3.2.2	Encryption	35
3.2.3	Stealth	37
3.2.4	Oligomorphism	38
3.2.5	Polymorphism	38
3.2.6	Metamorphism	46
3.2.7	Strong Encryption	47
3.3	Virus Kits	48
4.	ANTI-VIRUS TECHNIQUES	53
4.1	Detection: Static Methods	55
4.1.1	Scanners	55
4.1.2	Static Heuristics	69
4.1.3	Integrity Checkers	70
4.2	Detection: Dynamic Methods	71
4.2.1	Behavior Monitors/Blockers	71
4.2.2	Emulation	74
4.3	Comparison of Anti-Virus Detection Techniques	79
4.4	Verification, Quarantine, and Disinfection	80
4.4.1	Verification	81
4.4.2	Quarantine	82
4.4.3	Disinfection	82
4.5	Virus Databases and Virus Description Languages	85
4.6	Short Subjects	88
4.6.1	Anti-Stealth Techniques	88
4.6.2	Macro Virus Detection	89
4.6.3	Compiler Optimization	90

5. ANTI-ANTI-VIRUS TECHNIQUES	97
5.1 Retroviruses	97
5.2 Entry Point Obfuscation	99
5.3 Anti-Emulation	99
5.3.1 Outlast	99
5.3.2 Outsmart	100
5.3.3 Overextend	100
5.4 Armoring	101
5.4.1 Anti-Debugging	101
5.4.2 Anti-Disassembly	103
5.5 Tunneling	105
5.6 Integrity Checker Attacks	106
5.7 Avoidance	106
6. WEAKNESSES EXPLOITED	109
6.1 Technical Weaknesses	109
6.1.1 Background	110
6.1.2 Buffer Overflows	113
6.1.3 Integer Overflows	123
6.1.4 Format String Vulnerabilities	125
6.1.5 Defenses	127
6.1.6 Finding Weaknesses	132
6.2 Human Weaknesses	134
6.2.1 Virus Hoaxes	136
7. WORMS	143
7.1 Worm History	144
7.1.1 Xerox PARC, c. 1982	144
7.1.2 The Internet Worm, November 1988	145
7.2 Propagation	148
7.2.1 Initial Seeding	149
7.2.2 Finding Targets	150
8. DEWORMING	157
8.1 Defense	158
8.1.1 User	158
8.1.2 Host	158
8.1.3 Perimeter	163
8.2 Capture and Containment	167

8.2.1	Honeypots	168
8.2.2	Reverse Firewalls	169
8.2.3	Throttling	170
8.3	Automatic Countermeasures	172
9.	“APPLICATIONS”	177
9.1	Benevolent Malware	177
9.2	Spam	178
9.3	Access-for-Sale Worms	179
9.4	Cryptovirology	181
9.5	Information Warfare	182
9.6	Cyberterrorism	185
10.	PEOPLE AND COMMUNITIES	189
10.1	Malware Authors	189
10.1.1	Who?	189
10.1.2	Why?	190
10.2	The Anti-Virus Community	191
10.2.1	Perceptions	192
10.2.2	Another Day in Paradise	192
10.2.3	Customer Demands	194
10.2.4	Engineering	195
10.2.5	Open Questions	196
11.	WHAT SHOULD WE DO?	201
	References	205
	Index	223

List of Figures

1.1	Worm propagation curve	5
1.2	Ideal propagation curves for attackers and defenders	5
2.1	VGrep operation	20
2.2	Timeline of events	22
3.1	Multiple boot sector infections	29
3.2	Prepending virus	31
3.3	Appending virus	31
3.4	Concept in action	34
3.5	Encrypted virus pseudocode	35
3.6	Fun with NTFS alternate data streams	39
3.7	Virus kit	49
3.8	Virus kit, the next generation	49
4.1	Virus detection outcomes	54
4.2	Aho-Corasick finite automaton and failure function	56
4.3	Aho-Corasick in operation	57
4.4	Trie building	58
4.5	Trie labeling	59
4.6	Pattern substring selection for Veldman's algorithm	61
4.7	Data structures for Veldman's algorithm	62
4.8	Wu-Manber hash tables	63
4.9	Wu-Manber searching	63
4.10	The EICAR test file	65
4.11	Static vs. dynamic	72
4.12	From execution trace to dynamic signatures	73
4.13	Herding goats	77

4.14	Disinfection using checksums	84
4.15	Problem with unencrypted virus databases	86
4.16	Example virus descriptions	88
5.1	Checking for single-stepping	102
5.2	False disassembly	103
5.3	Anti-disassembly using strong cryptographic hash functions	104
5.4	On-demand code decryption	105
6.1	Conceptual memory layout	110
6.2	Sample segment allocation	111
6.3	Stack frame trace	112
6.4	Before and after a subroutine call	113
6.5	Code awaiting a stack smash	114
6.6	Stack smashing attack	115
6.7	Environmentally-friendly stack smashing	116
6.8	Code that goes just a little too far	117
6.9	Frame pointer overwrite attack	118
6.10	A normal function call with arguments	119
6.11	Return-to-library attack, with arguments	120
6.12	Overflowing the heap onto bookkeeping information	121
6.13	Dynamic memory allocator's free list	121
6.14	Normal free list unlinking	122
6.15	Attacked free list unlinking	123
6.16	Code with an integer overflow problem	124
6.17	Stack layout for calling a format function	126
6.18	Code with a format string vulnerability	127
6.19	Format string attack in progress	128
6.20	Canary placement	130
6.21	"It Takes Guts to Say 'Jesus'" virus hoax	136
6.22	"jdbgmgr.exe" virus hoax	137
7.1	A conversation with <code>sendmail</code>	146
7.2	Finger output	146
7.3	TCP connection establishment	148
7.4	IP address partitioning	150
7.5	Permutation scanning	152
8.1	An example network	157
8.2	Rate of patching over time	159

8.3	Signatures in network traffic	165
8.4	Traffic accepted by an IDS and a host	166
8.5	TTL attack on an IDS	167
8.6	Network traffic throttling	171
9.1	Organized crime and access-for-sale worms	180
9.2	Disorganized crime and access-for-sale worms	180
10.1	Malware analysis workflow	193
10.2	In the zoo vs. in the wild	195

Preface

It seemed like a good idea at the time. In 2003, I started teaching a course on computer viruses and malicious software to senior undergraduate and graduate students at the University of Calgary. It's been an interesting few years. Computer viruses are a controversial and taboo topic, despite having such a huge impact on our society; needless to say, there was some backlash about this course from outside the University.

One of my initial practical concerns was whether or not I could find enough detailed material to teach a 13-week course at this level. There were some books on the topic, but (with all due respect to the authors of those books) there were none that were suitable for use as a textbook.

I was more surprised to find out that there was a lot of information about viruses and doing "bad" things, but there was very little information about anti-virus software. A few quality minutes with your favorite web search engine will yield virus writing tutorials, virus source code, and virus creation toolkits. In contrast, although it's comprised of some extremely nice people, the anti-virus community tends to be very industry-driven and insular, and isn't in the habit of giving out its secrets. Unless you know where to look.

Several years, a shelf full of books, and a foot-high stack of printouts later, I've ferreted out a lot of detailed material which I've assembled in this book. It's a strange type of research for a computer scientist, and I'm sure that my academic colleagues would cringe at some of the sources that I've had to use. Virus writers don't tend to publish in peer-reviewed academic journals, and anti-virus companies don't want to tip their hand. I would tend to characterize this detective work more like historical research than standard computer science research: your sources are limited, so you try and authenticate them; you piece a sentence in one document together with a sentence in another document, and you're able to make a useful connection. It's painstaking and often frustrating.

Technical information goes out of date very quickly, and in writing this book I've tried to focus on the concepts more than details. My hope is that the

concepts will still be useful years from now, long after the minute details of operating systems and programming languages have changed. Having said that, I've included detail where it's absolutely necessary to explain what's going on, and used specific examples of viruses and malicious software where it's useful to establish precedents for certain techniques. Depending on why you're reading this, a book with more concrete details might be a good complement to this material.

Similarly, if you're using this as a textbook, I would suggest supplementing it with details of the latest and greatest malicious software that's making the rounds. Unfortunately there will be plenty of examples to choose from. In my virus course, I also have a large segment devoted to the law and ethics surrounding malicious software, which I haven't incorporated here – law is constantly changing and being reinterpreted, and there are already many excellent sources on ethics. Law and ethics are very important topics for any computer professional, but they are especially critical for creating a secure environment in which to work with malicious software.

I should point out that I've only used information from public sources to write this book. I've deliberately excluded any information that's been told to me in private conversations, and I'm not revealing anyone's trade secrets that they haven't already given away themselves.

I'd like to thank the students I've taught in my virus course, who pushed me with their excellent questions, and showed much patience as I was organizing all this material into some semi-coherent form. Thanks too to those in the anti-virus community who kept an open mind. I'd also like to thank the people who read drafts of this book: Jörg Denzinger, Richard Ford, Sarah Gordon, Shannon Jaeger, Cliff Marcellus, Jim Uhl, James Wolfe, and Mike Zastre. Their suggestions and comments helped improve the book as well as encourage me. Finally, Alan Aycock suggested some references for Chapter 10, Stefania Bertazon answered my questions about rational economics, Moustafa Hammad provided an Arabic translation, and Maryam Mehri Dehnavi translated some Persian text for me. Of course, any errors that remain are my own.

JOHN AYCOCK