

Spyware and Adware

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.

For a complete list of titles published in this series, go to www.springer.com/series/5576

John Aycock

Spyware and Adware

 Springer

John Aycock
Department of Computer Science
University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4
aycock@ucalgary.ca

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.

ISSN 1568-2633
ISBN 978-0-387-77740-5 e-ISBN 978-0-387-77741-2
DOI 10.1007/978-0-387-77741-2
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010935807

© Springer Science+Business Media, LLC 2011

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 on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

for melissa and amanda

Contents

1	Introduction	1
1.1	Definitions and History	1
1.2	Motivation	4
2	Getting There	9
2.1	Installation	9
2.1.1	Explicit, Voluntary Installation	9
2.1.2	Drive-by Downloads, User Involvement	10
2.1.3	Drive-by Downloads, No User Involvement	16
2.1.4	Installation via Malware	19
2.2	Startup	20
2.2.1	Application-Specific Startup	20
2.2.2	GUI Startup	21
2.2.3	System Startup	22
2.2.4	Kernel Startup	22
2.2.5	Defenses	23
3	Staying There	29
3.1	Avoiding Detection	29
3.1.1	Basic Detection Avoidance	29
3.1.2	Anti-Spyware	32
3.1.3	Advanced Detection Avoidance: Rootkits	33
3.2	Avoiding Uninstall	37
3.2.1	Passive Avoidance	37
3.2.2	Active Avoidance	38
4	Keylogging	45
4.1	User Space Keylogging	47
4.1.1	Polling	47
4.1.2	Event Copying	48
4.1.3	Event Monitoring	48

4.2	User Space Keylogging Defenses	49
4.3	Authentication	53
5	Phoning Home	59
5.1	Push vs. Pull	59
5.2	Finding Home	61
5.3	Steganography	63
5.4	Information Leaking Defenses	66
6	Advertising	71
6.1	Types of Advertisement	71
6.1.1	Banner Advertisement	74
6.1.2	Banner Advertisement with Pull-down Menu	75
6.1.3	Expandable Banner Advertisement	76
6.1.4	Pushdown Banner Advertisement	77
6.1.5	Pop-up Advertisement	77
6.1.6	Pop-under Advertisement	78
6.1.7	Floating Advertisement	79
6.1.8	Tear-back Advertisement	79
6.1.9	In-text Advertisement	80
6.1.10	Transition Advertisement	81
6.1.11	Video Advertisements	82
6.2	Intent and Content	83
7	Advertisement Implementation	91
7.1	Implementation Location	92
7.1.1	Implementation on the User Machine	92
7.1.2	Implementation in the Network	96
7.1.3	Implementation near the User Machine	97
7.1.4	Implementation on the Server	98
7.2	Choosing Keywords	99
7.3	Blocking Advertisements	101
7.3.1	Pop-up Blocking	101
7.3.2	General Advertisement Blocking	102
7.3.3	Blocker Evasion and Blocker Blocking	103
8	Tracking Users	111
8.1	Cookies	111
8.1.1	Defenses	116
8.1.2	Other Browser-Related Tracking Methods	117
8.2	User Profiling	118
8.2.1	Cognitive Styles, Mood, and Personality	119
8.2.2	Future Actions	119
8.2.3	Demographic Information	120
8.2.4	Social Networks	120
8.2.5	Real World Activities	121

8.2.6	Physical Location	121
8.2.7	Search Terms and Keywords	122
8.2.8	Disinterests	122
9	Conclusion	127
	References	129
	Index	143

List of Figures

2.1	End-user license agreement excerpt	10
2.2	Software installation prompt	11
2.3	Alice asymmetrically encrypts a message to Bob	12
2.4	Alice signs and sends some code to Bob	13
2.5	A deceptive software name	14
2.6	Part drive-by, part voluntary installation	15
2.7	Normal execution	17
2.8	Stack smashing attack	17
2.9	Attack string for stack smashing attack	18
3.1	Executing encrypted spyware	30
3.2	Code mutation	31
3.3	Normal flow of information	34
3.4	Hooking shared library functions	35
3.5	System call hooking in the kernel	36
4.1	Password-stealing opportunities	45
4.2	Pseudocode for a polling keylogger	48
4.3	Pseudocode for an event-copying keylogger	48
4.4	Pseudocode for an event-monitoring keylogger	49
4.5	Menu-based password entry	51
4.6	Virtual keyboard password entry	51
4.7	Virtual keyboard capture with partial image	52
4.8	Animated symbols as a screen shot defense	52
4.9	Changing symbol layout for each password entry	52
4.10	Selecting characters by mouse hovering	53
4.11	Virtual mouse pointer	53
4.12	Two-factor authentication	55
5.1	Sample hosts file	61
5.2	Fast flux with proxies and mother ship	62

5.3	Web page with steganographic message	63
5.4	PPM file without embedded message	64
5.5	PPM file with embedded message	65
5.6	Exfiltration using ICMP echo	66
6.1	Interstitial or not?	72
6.2	Trivial user interaction	73
6.3	Banner advertisement	74
6.4	Banner advertisement located beside content	75
6.5	Banner with pull-down menu	76
6.6	Expandable banner	76
6.7	Pushdown banner	77
6.8	Pop-up advertisement	78
6.9	Pop-under advertisement	79
6.10	Floating advertisement	80
6.11	Tear-back advertisement	80
6.12	In-text advertisement	81
6.13	Transition advertisement	82
6.14	Non-linear video advertisements	83
7.1	Floating box implementation	92
7.2	Locations for implementing advertisements	93
7.3	Centralized advertising software	94
7.4	Typhoid adware	97
8.1	HTTP transaction with cookies	112
8.2	Cookies in detail: multiple HTTP transactions	114
8.3	Fetching third-party content	114
8.4	Fetching third-party content, with cookies	115
8.5	Tracking user browsing over multiple web sites	115
8.6	Tracking using Cascading Style Sheets	117

Preface

It was a dark and stormy night.

Actually, I don't remember now. What I *do* remember is that in November 2004, I sent a lone email to my department head at the University of Calgary, with a carefully-worded question: what is the department's tolerance for potentially controversial courses?

There was some historical precedent for that precise wording. I had sent him a similar email message early in 2003 as a prelude to starting my course on computer viruses and malware [26]. That course was one of a handful in the world, and I believe the only one in Canada at the time, to take a "hands-on" approach to computer viruses, where students created their own viruses and anti-virus software in a secure laboratory environment [25].

Fast-forward to 2005. Spam and spyware, the course initiated by my innocent-looking 2004 email, makes its debut [24]. It also was hands-on, and was and is, to the best of my knowledge, the only course of its kind in the world. I wish I would be proven wrong on this claim, because I think that both are important topics that should be taught to computer science students – after all, these students are the next generation of Internet defenders.

One problem I had teaching this course was the lack of good textbooks, for spyware in particular. Even in 2010, four offerings of the course later, there is still no contender. The information *is* out there, though, and this book is the result of my efforts to gather all this information together and organize it in some meaningful way.

There are three things that have been deliberately excluded from this book. First, I spend time in my class teaching about spyware-related legal aspects, and I have included none of this. The laws regarding spyware are still in flux currently, and in any case are jurisdiction-specific. Second, there is also ethics content relating to spyware in my course, but there are lots and lots of good ethics books already. Third, I am excluding certainty. While it would be great to say that spyware always does *this* and spyware never does *that*, it would be very foolish to do so. Spyware is software that can be made to do an infinite number of things, in an infinite number of ways. Instead, except when specific examples are discussed, I will stick to the *cans*

and *mays* and *coulds* and *mights* that suggest the full scary potential of spyware. There are few certainties in malicious software, sorry.

I have avoided using code (except pseudocode) as much as possible in this book. The ideas and concepts are the most important things here, and I assume that the reader has enough programming experience to determine implementation specifics. Also, code tends to give books the same shelf life as a loaf of bread. I'd prefer to avoid that. Some knowledge of operating systems and networking is also useful, although I try to explain more esoteric points as needed.

Some words of caution: implementation and/or use of some techniques described in this book may not be legal in the reader's part of the world. This information is not provided to help the "bad guys," who probably already know all this anyway, but facilitate the training of the "good guys." Also note that some techniques are covered by patents. While I have made attempts to cite relevant patents when possible, their language can be very broad in scope, and it is almost certain that I have inadvertently missed some. Citations to patent applications and assigned patents are for reference purposes only and are not meant to endorse the validity of their claims.

On the topic of references, each chapter has notes that contain citations, s(n)ide comments, and extra information. To avoid disrupting the flow of the text when reading, the margins contain small circles indicating the lines that have associated notes.

I would like to thank Ken Barker and the Department of Computer Science for supporting this course to begin with. Although the details are several levels above my pay grade, I probably also owe thanks to more senior administrative people at the University of Calgary for backing my security courses too. Many thanks to all the students that have taken the course; their questions helped keep me on my toes. This book was proofread and commented on in whole or part by Angelo Borsotti, Heather Crawford, Jörg Denzinger, Shannon Jaeger, Jim Uhl, and Mike Zastre. Heather Crawford and James Ong pointed me to some helpful references, Philip Fong answered my questions about information flow control, and Jason Franklin clarified a point about a paper of his. Their collective advice has hopefully kept my details correct and my modifiers from dangling.

John Aycock