Software Forensics

Collecting Evidence from the Scene of a Digital Crime

Robert M. Slade

McGraw-Hill

New York Chicago San Francisco Lisbon London Madrid
Mexico City Milan New Delhi San Juan Seoul
Singapore Sydney Toronto

The McGraw-Hill Companies

Library of Congress Cataloging-in-Publication Data on file

Copyright © 2004 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 0 1 9 8 7 6 5 4

ISBN 0-07-142804-6

The sponsoring editor for this book was Judy Bass and the production supervisor was Pamela A. Pelton. It was set in Sabon by Patricia Wallenburg. The art director for the cover was Anthony Landi.

Printed and bound by RR Donnelley.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Director of Special Sales, McGraw-Hill Professional, Two Penn Plaza, New York, NY 10121-2298. Or contact your local bookstore.

This book was printed on recycled, acid-free paper containing a minimum of 50% recycled, de-inked fiber.

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. ("McGraw-Hill") from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein, and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

2

The Players—Hackers, Crackers, Phreaks, and Other Doodz

Because we may be using software forensics to attempt to identify authors of software, it may help to have a rough idea of the type of people we are looking for. Those who write malicious software, or attempt to distribute or resell commonly available commercial software, tend to belong to communities of like-minded individuals. Over the years, we have been able to glean ideas about the characteristics of this tribe. For this information, we are all indebted to researchers such as Sarah Gordon, Dorothy Denning, Ray Kaplan, and, more recently, the members of the Honeynet Project.

A couple of provisos: Whenever you deal with people, there will always be exceptions. There are those who seem to pursue security breaking from motives that are, if not exactly admirable, at least untainted by thoughts of commerce or attention. Indeed, we can't really say that all endeavors related to the creation of viral software or intrusion utilities are even illegal. While most of the activity involved in security breaking is highly repetitive, there are also those few who do come up with one or two original ideas, and experiment with them.

As another example of a deviation from a stereotype, most studies of those involved in security breaking activities have been done

in western societies: Europe, North America, and Australia. Recently, groups have been quite visible in China. There are two major populations, the red guests, and the black, or terrible, guests. The black guests are apparently quite akin to Western groups, with a lack of cooperation, antiestablishment positions, and random activities. The red guests, on the other hand, seem to form very stable groups, are executives in technology companies, have links with the Chinese government, and run coordinated exercises. In this case, we have a very large group running completely contrary to the expected norms for the community, and this may be derived from the differing foundations of Eastern and Western social thought.

Therefore, we can't make blanket statements about all of those within such a community. However, as with almost any stereotypes, there are reasons for the characterizations presented here.

Particularly in doing forensic analysis, we need to beware of falling into mental traps occasioned by our own "profiles" of the adversary. If we get too caught up in any one idea, we are going to blind ourselves to important evidence, whether it be proof of innocence or guilt. While it is beneficial to have an idea of the attributes of the majority of the people we are studying, it is absolutely vital always to accept the possibility of exceptions.

Terminology

When dealing with the blackhat communities and products, and malicious software in particular, there is a good deal of specialized jargon that does have meaning, but tends to be thrown around rather carelessly. Please bear with me in this section, as I will be mentioning some of it before it gets rigorously defined. By the end of the chapter, all should be clearly revealed.

The perceptive may have noted that I have not, except in the title, used the term "hacker." This is because there is considerable controversy in regard to the use of the word. Originally, the term meant one who was skilled in the use of computer systems, particularly if that skill was acquired in an exploratory manner. The usage applied to all aspects of the technology, whether hardware or software. In fact, it came to be extended to all forms of expertise: A hacker was a master of his (or her) craft, and the term was roughly equivalent to wizard or guru. Those who pursued this level of proficiency were usually those who were more than a little obsessed

with it and therefore considered what the rest of the world sees as social skills to be, at best, inconsequential. Therefore, they gained a reputation for being uncommunicative and disdainful of notions of property and propriety, despite the fact that various forms of "hacker ethic" generally promoted the education of others and injunctions not to damage systems or data.

Later, the term came to be applied, usually by the media, to skilled or unskilled who break security systems. Originally, there may have been some merit in this usage. When the ability to communicate with computers at all was an arcane art, proficiency in connecting to them without authorization and getting them to perform for you was only acquired with patient investigation. However, as modems became more common, and as tricks for getting around access controls were distributed through bulletin boards, the level of skill required dropped significantly. It is easy to see why those who were trying to break into computers encouraged people to call them hackers: They assumed a mantle of mastery and superiority by virtue of a limited, though not really special, knowledge. In fact, a number of the members of the community came to be known as "wannabes" in their attempt to "want-to-be" seen as possessing skills that they did not, in reality, have.

Actually, you can determine a person's level of technical expertise by how he uses the term. Someone who uses hacker as meaning an expert is someone who generally does advanced technical work. Someone who uses hacker as a "bad guy" may have a technical background of some type, or a technical job, but usually is nowhere near the cutting edge.

So what do we call people who are breaking into computers or writing malicious software? An attempt was made some years ago to rehabilitate the term hacker, and to call those who tried to break, or crack, security systems, "crackers." Unfortunately, this attempt never did succeed with the general public, and there is a problem of confusion with those who break anticopying technologies on commercial software, who are also known as crackers.

In an attempt to avoid debates about "good" hackers versus "bad" hackers versus "crackers" versus "phone phreaks" versus "virus writers" versus "vxers" (and we will discuss the segmentation of the dark side population shortly), the security community has taken to describing those who either attempt to break into computer systems without prior authorization, or who explore security

primarily from an attack perspective, as "blackhats." The term originates from the genre of old American western movies where the "good guys" always wore white hats and the "bad guys" always wore black. By a fairly automatic extension, those who attempt to explore security solely from the perspective of defense are the whitehats. (And, of course, with the world of computer security being convoluted, anyone who seems to sail fairly close to the line is known as a grayhat.)

Once again, I need to repeat my earlier point in regard to assuming too much. The term "blackhat" is a label of convenience for describing a broad class of activities and individuals. Not all people involved in blackhat groups are performing illegal activities. A series of security seminars has taken to using the term "Blackhat Conference." In regard to hiring those who have done computer intrusions to perform penetration tests of security, there is now discussion of "ethical blackhats." Therefore, it is safest to bear in mind that the term is most frequently used in regard to a perspective on systems and security, and to avoid dealing with moral judgments at this point.

Types of Blackhats

The blackhat community is extremely fragmented. Not only are there different groups, often at odds with each other, but the types of activities also differ. Despite the omnicompetent evil genii portrayed in fiction about "hackers," there is a great deal of specialization in the real blackhat groups, and those from one clique seldom do much exploration in the other fields.

Some are trying to break into or intrude on computer systems or networks. These are the ones who most frequently are given the hacker sobriquet, and are usually referred to as "crackers" (or system crackers, to distinguish them from the software piracy-type crackers) by the security community. Despite the general public reputation, few of these people do any programming, or create any sort of software, malicious or otherwise. There are a limited number of system crackers who do analyze software, and particularly system software, for weaknesses, and who then write exploit tools to automate the process of breaking in. However, these tools are, generally speaking, not a major problem. They are specific to a given system and version, and, even if distributed and utilized, have a very limit-

ed lifespan. If a particular vulnerability is widely exploited, then it tends to become known and patched quickly.

Other blackhats specialize in gaining unauthorized use of telephone switches and systems, usually for their own aims and amusement, but possibly for the purpose of obtaining or even reselling phone service. Those interested in breaking into or otherwise manipulating the telephone system are referred to (and refer to themselves) as "phone phreaks," using the punning variant spelling. This is generally shortened to "phreaks" in common usage. (Variant spelling, and even the use of nonalphabetic characters, is a characteristic of most blackhat communities. The effect is to define the population of the group, separating those who know the jargon, and therefore belong, from those who do not. Thus, those within can see themselves as members of an elite club—but probably represent it as "leet" or "3!33t." Hence also the reference to "doodz" [dudes] in the title of this chapter.) The act of manipulating the phone system is known as "phreaking."

Some are primarily interested in damaging or corrupting files, particularly in public ways, such as defacing Web sites. This runs

HACKTIVISM

Hacktivism is a convenient label, but a poorly defined term. Hacktivism can be anything that the user, generally a journalist, defines. It can be writing a new utility and releasing the same with attached political or social advertising. It can be developing a new Web site to promote civil rights or social change. It can also be developing online direct actions against corporations or governments, through mechanisms using the Internet.

The Internet enables debate or action on many issues. When we understand how people are using this new medium, we see a variety of social activities beyond online shopping and swapping pictures of family pets. For the online activists, such as the "electrohippies," understanding how different groups perceive the Internet is the first step in comprehending that these groups feel they are developing, or influencing, a new online consciousness that can create a new environment for realizing societal change, potentially globally.

contrary to versions of the "hacker ethic," because most of the documents identified as such contain some kind of "do no harm" provisions. However, many system crackers operate primarily on an ego drive, and need to have some way to prove an intrusion and keep score. In addition, a more recent attempt to prove the value of system breaking as a means of social protest, known as "hacktivism," uses the defacement of targeted Web sites as a vehicle for publicity and activism.

At its root, hacktivism is seeking to use advanced knowledge of IT systems to change the way people use and relate to computer hardware or computer networks, as well as each other.

A great many of the blackhats in general, and probably the largest majority, really have very little idea of the technology that they are using, having obtained packaged programs or scripts, and they are operating them without really understanding the functions or situations appropriate for their use. The vast majority of intrusion activity on the Internet arises from these "script kiddies" who have obtained utilities and scripts and simply launch attacks against random addresses. It is difficult to say that such attacks are even malicious. They are certainly thoughtless, and the time and resources necessary to deal with them are a drain on the resources of both institutions and individuals.

Those who create programs of any type, whether utilities or malware, are actually relatively rare. A number do make slight modifications to the creations of others, usually functionally insignificant changes to viruses, which are widely available because of their reproductive function. Thus, there are the vast, and usually closely related virus "families," and the phenomenon that when a new type of exploit tool arrives on the scene, it is quickly followed by a half-dozen extremely similar programs. The inordinately repetitive and noncreative nature of most of this programmed material may explain the contempt in which virus writers are held by large numbers of other blackhats. The production of viruses is seen, correctly, as a rather trivial exercise, rather than proof of programming skill.

There are, of course, those who are preoccupied in making illegal copies of commercial software. The "warez doodz" are generally most interested in collecting, and sometimes redistributing, such packages. A few, though, specialize in the analysis necessary to break systems designed to prevent just such copying. In some cases,

these crackers also produce software dedicated to automating the copying or registering of commercial software.

And, at every level, there are those who "wannabe" more respected in the blackhat community, but lack even those skills.

Motivations and Rationales

It may be important to examine the commonly presented justifications for blackhat activity. There are two reasons for this study. First, this examination demonstrates something of the mindset and philosophy of the members of the community, and such a philosophy can sometimes be evident in programming style. Second, some of these justifications may be presented, quite seriously, as arguments against the activity of software forensics in general.

One of the most frequently attempted justifications of blackhat activity of all kinds is that it is protected under the concept of freedom of speech. Leaving aside the issue of whether free speech is a universal right, we also have to ignore for the moment the fact that most blackhat activity does not involve programming. In addition, we still have to ask whether programming is, or is not, speech. Speech generally does not involve other people, and when it does, such as in the case of yelling "Fire!" in a crowded theater or producing hate propaganda, it often is not protected. Programs may be used to express a message or idea of some sort, even beyond the text that such a system may carry or present. However, the determination of whether code actually constitutes speech can be extremely difficult, and has been decided both ways when presented before the courts. The concept of "artistic merit," which is usually considered in such cases, is unlikely to support the blackhat argument in terms of its usual products. In the case where the blackhat individual is not the author of the software, such as where attack scripts are being utilized or pre-existing viruses are being released, the protection of free speech is even more tenuous.

The freedom of speech argument resonates strongly, not only with society at large, but particularly with the blackhat community. Noting the self-identification with the original hackers, we frequently see statements such as "information wants to be free." There is a large measure of curiosity in the blackhat community, as an important characteristic. Among those who have been charged with computer trespass, a frequent claim is that the intruder "only

wanted to know." While far from any justification, this motivation is probably true in a great many cases. The search for additional purposes is likely a waste of effort and a distraction from analysis of the real situation.

A second bid at vindication of security breaking activities is simply "because we can." Although the shallowness of this argument tends to prompt a sarcastic response from security or law enforcement personnel, we should note that the prevalence of this reasoning does make a very strong point about the anarchic nature and mindset of the blackhat community.

The idea of freedom itself is an important one. Note the competitive nature and divisiveness of the blackhat population overall. Note the tendency of blackhats to be loners and undersocialized. Freedom, in its most anarchic form, is an attribute of blackhats. Cooperation between individuals is rare, and between groups, exceptional. Therefore, evidence of multiple authors is frequently also an indication that code has been copied from another source.

Many individuals who practice system violation activity explain themselves on the basis that they are following in the footsteps of the old-time hackers, who explored and discovered the capabilities of early computing devices. This flies in the face of the reality of the current level of blackhat endeavors: The few instances that are not absolutely repetitive are generally slavishly derivative. Even if we ignore the fact that most "cracking" exercises amount to no more than "knocking on doors," we still have to ask what the objective of these explorations is, which usually cannot be clearly articulated, and look at the eventual result, which has not, to date, been anything significant.

However, in addition to the curiosity factor noted earlier, this does point out an important trait in blackhats. Ego drive is an extremely strong motivation. Therefore, we do not have to look for additional reasons, such as a profit motive, to explain activities. "I can do it," is quite sufficient. Searching for an idea of who would profit from a given operation is probably fruitless.

Yet another justification for blackhat activities is stated to be educational. As one who has been involved in education and training, as well as reviewing, for a great many years, I would be very sympathetic to this argument—if it had any basis. Even considering *2600* magazine, which can most charitably be described as the best of a bad lot, one is hard pressed to say anything positive about the

writing quality, research, originality, or even such basics as sticking to the topic. When one turns to *phrack*, *40Hex*, and the myriad others of the "zine" ilk, the caliber runs steadily downhill. Even articles addressing simple penetration testing generally state only that systems are weak (we already knew that, thanks), and say nothing about strengthening them.

The educational activities, therefore, tend to be rather thinly veiled boasting. This characteristic may render all study of software forensics moot: Most individuals convicted of malware or security penetration offenses have been caught because of their own statements.

Hacker Manifestos and Other Documents

Please note that the texts reprinted here have been formatted for line length, but are otherwise unedited, and contain the original errors in grammar and spelling.

Almost from the invention of the computer there have been documents describing the characteristics and behavior of the skilled operator. The gist of a number of these has been compiled in the collection of materials known as "The Jargon File." This is an excellent source for gauging the mindset of those truly skilled with computers, and the type of position to which most individuals in the blackhat communities aspire. "The Jargon File" can be found at any number of Web sites, including http://www.elsewhere.org/jargon/html/index.html or http://info.astrian.net/jargon/.

From "The Jargon File," the entry on hacker ethics reads:

> "hacker ethic *n.* 1. The belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing open-source and facilitating access to information and to computing resources wherever possible. 2. The belief that system-cracking for fun and exploration is ethically OK as long as the cracker commits no theft, vandalism, or breach of confidentiality."

In writings such as "Old and New Hacker Ethics," at http://cgi.fiu.edu/~mizrachs/hackethic.html, this is expanded to include items such as:

- Access to computers and hardware should be complete and total. (This is usually known as the hands-on imperative.)
- Information wants to be free. (This assertion has become almost a mantra and item of faith among various segments of the information technology world, both within and outside the blackhat community.)
- Mistrust authority and promote decentralization.
- Hackers should be judged by their hacking, not by irrelevant criteria such as race, age, sex, or position.
- You can create truth and beauty on a computer.
- Computers can change your life for the better.

Another set, found at http://www.crackinguniversity2000.it/alt-hacking-FAQ/15.html, includes the following points:

"Do not destroy/damage files unless it is absolutely necessary to cover your traces

"Notify the system administrator about any holes in security that you have found/exploited

"Patch any holes in security that you have found/exploited

"Do not break into a system for money. Do not steal money. Don't distribute information/software for money.

"Treat a machine that you break into as you would treat your own.

"Document/spread what you have learned"

Other prescriptive documents, outlining a level of moral activity, are similar to the "hacker ethic" found at http://www.data-sync.com/~sotmesc/etc/ethics.html that reads as follows:

> "Hacking has Ethics you ask? Of course, though the media makes it seem like we are criminals, only a few of us are. I true hacker lives to know. A true hacker does not break into a system and delete its file system, plant and run viruses or try to destroy the data within. These however, seem to be the most known characterics of the "New Generation." These newbies, who most likely got a computer and Internet as a present, and only know the basic of whatever OS their system came with have a lot to learn.
>
> "They see a program, and they weigh the trouble of learning to use it well over how destructive it is. True

hackers use anonymous mail to cloak themselves, not send mail bombs. True hackers do not use Winnuke, or anyother DoS attack, unless it is to gain access to a server/network or in a act of self defense. Some of the newbies out there today, get a program like Winnuke simply to see how many people they can take out, if they'd research it a little more, they'd find out anyone with a patch is more then likely immune to Winnuke.

"True hackers, after the first week or so, release that if they keep asking for help, without even trying to find the answer will get flamed. These newbies, when finally gaining access to a system try to take it out. They try to employ programs like BO or Netbus, simply so they can use the term "Hacker" on there name. They are Wrong! More then likely they don't understand how the program even works, evidence alone on any hacking msg board, newbies asking questions without reading.

"Granted, I do not believe I should be called a true hacker yet. yet I know the difference between hacking and crashing, something more then likely the newbies will do. That or become a Warez Pirate, so they can be "k00l" or "313373" they have yet to understand what a hacker truly is, and most likely never will."

Currently, the most famous of the hacker credos is the "Hacker's Manifesto," written in 1986 by Loyd Blankenship under the pseudonym of The Mentor. It is reproduced in various forms around the World Wide Web, at sites such as http://manifestopost.com/famous/mentor.html or http://www.humanunderground.com/archive/hackermanifesto.html.

The text generally reads similarly to the following:

"Another one got caught today, it's all over the papers. 'Teenager Arrested in Computer Crime Scandal', 'Hacker Arrested after Bank Tampering.' 'Damn kids. They're all alike.' But did you, in your three-piece psychology and 1950s technobrain, ever take a look behind the eyes of the hacker? Did you ever wonder what made him tick, what forces shaped him, what may have molded him? I am a hacker, enter my world. Mine is a world that begins

with school. I'm smarter than most of the other kids, this crap they teach us bores me. 'Damn underachiever. They're all alike.' I'm in junior high or high school. I've listened to teachers explain for the fifteenth time how to reduce a fraction. I understand it. 'No, Ms. Smith, I didn't show my work. I did it in my head.' 'Damn kid. Probably copied it. They're all alike.' I made a discovery today. I found a computer. Wait a second, this is cool. It does what I want it to. If it makes a mistake, it's because I screwed it up. Not because it doesn't like me, or feels threatened by me, or thinks I'm a smart ass, or doesn't like teaching and shouldn't be here. Damn kid. All he does is play games. They're all alike. And then it happened… a door opened to a world… rushing through the phone line like heroin through an addict's veins, an electronic pulse is sent out, a refuge from the day-to-day incompetencies is sought… a board is found. 'This is it… this is where I belong…' I know everyone here… even if I've never met them, never talked to them, may never hear from them again… I know you all… Damn kid. Tying up the phone line again. They're all alike… You bet your ass we're all alike… we've been spoon-fed baby food at school when we hungered for steak… the bits of meat that you did let slip through were prechewed and tasteless. We've been dominated by sadists, or ignored by the apathetic. The few that had something to teach found us willing pupils, but those few are like drops of water in the desert. This is our world now… the world of the electron and the switch, the beauty of the baud. We make use of a service already existing without paying for what could be dirt-cheap if it wasn't run by profiteering gluttons, and you call us criminals. We explore… and you call us criminals. We seek after knowledge… and you call us criminals. We exist without skin color, without nationality, without religious bias… and you call us criminals. You build atomic bombs, you wage wars, you murder, cheat, and lie to us and try to make us believe it's for our own good, yet we're the criminals. Yes, I am a criminal. My crime is that of

curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for. I am a hacker, and this is my manifesto. You may stop this individual, but you can't stop us all... after all, we're all alike."

General Characteristics

Blackhats, and particularly writers of malware and viruses, tend to be young, and almost invariably male. Despite occasional speculations on the addictive nature of "hacking," they usually "grow out of" the virus writing game after a few years. In terms of forensics, then, you are unlikely to find that the same author has continuously modified the same piece of malware over many years.

Why is the typical malware author young? Writers of fiction, and those in the media, tend to infer from this fact that technology is a young person's game, and that the elderly—those of the venerable age of say, 25—have lost the necessary mental acuity to produce viral code. When looking at the reality, however, this is far from the truth. With age comes experience, skill, and a library of past work. Therefore, programmers tend to become more productive in terms of the time taken to produce a given piece of code. Time is, though, an important factor. With age also comes a greater number of responsibilities and demands on time. Technical managers know that young programmers can be counted on to "pull all-nighters": older coders develop a slightly differing view of the importance of the work they are being asked to do.

In fact, malware authors are definitely those of whom it can be said that they should "get a life." Writing malicious code is not particularly difficult or sophisticated, but it does take a lot of time. Therefore, those involved in the practice are people who have nothing better to do. (In his thought-provoking essay, "Losing Your Voice on the Internet," James DiGiovanna points out that while virus writing is an attention-seeking behavior, it is inherently futile because the author's identity is seldom known to the general public, and the only importance of the virus itself is that it be eliminated.) In virus research, we have noted that virus authors almost univer-

sally mature beyond the malware game within a few years. The fact that most malware authors are also male has unfortunate implications for the importance of male, as opposed to female, roles in most societies.

Antivirus researchers tend to be dismissive of the technical abilities of virus writers. There are virus writers who write competent code; there are many more who do not. As noted earlier, the vast majority of malicious code is copied from earlier examples, with only very minor modifications. What variation is made tends to be cosmetic rather than functional: Often malware authors understand so little of the software they are working with that they dare not make changes to operational sections. In addition, malware is riddled with bugs of all types, indicating both carelessness and a general lack of skill.

The industry's lack of respect for the abilities of virus writers is well counterbalanced by the media, who continue to be fascinated by the mythical boy genius running rings round the incompetent antivirus workers. Malware authors like this cliché, too, and go to some lengths to encourage the stereotype.

Most of today's malware programmers gain access to a victim system by tricking the victim into executing malicious code. It is much easier to fool people than to identify possible exploits and find ways to effectively use them. Therefore, a preference for the "easiest" option is quite characteristic of malware programming.

Malware writers don't understand or prefer not to think about the consequences for other people, or they simply don't care. Recently, one researcher has speculated on the characteristics of the blackhat community in comparison to those of people who fall somewhere in the range between an admittedly ill-defined "normal" and those suffering from full-blown autism. Autistic individuals tend to perceive and interpret the world in an idiosyncratic manner.

Malware authors draw a false distinction between creating malicious software and distributing it. They eschew any responsibility for the damage caused by their creations. In particular, they believe it is the responsibility of the victim to defend him or herself from encroaching malware, not the responsibility of the creators to keep their handiwork away from systems other than their own. Targets and victims of attacks are typically dehumanized in blackhat writings, described as losers who do not deserve to own a computer. There is also projection and displacement of guilt, frequently

expressed in terms justifying security breaking activities because a certain vendor makes poor quality software or because large corporations are doing bad things.

In self-reports from blackhats, a number of aspects are reported to be part of the thrill, including the act of vandalism itself, fighting authority, "matching wits" with the security or law enforcement communities, aggression (often arising out of resentment, and reinforced by the feeling of safety and power that is engendered by apparent anonymity), the ability to induce fear and panic in the media and the general public, and the "15 minutes of fame" as well as the recognition of peers. Malware writers tend to feel marginalized and unrecognized in normal society, so they feel a very strong sense of identity with the blackhat tribe, even while denigrating other members of that same community.

Blackhat Products

Most of the end result of blackhat activity consists of compromised systems, defaced Web pages, and pointlessly consumed bandwidth. Overall, this might be of interest to those investigating network forensics, but isn't of much use for us in software forensics. However, attack tools, distributed denial of service (DDoS) kits, trojans, viruses, worms, remote access trojans (RATs), and other forms of malware are.

We will, of course, want to find out as much as possible about what the specific piece of malware does. We also want to know about the author, if we can. Becoming familiar with the broad classes of malicious software can help point out, in general outline, the functions to look for. Knowing the class of malware may also help us to identify the author, because blackhats tend to be just as specialized as any other type of programmer.

It is sometimes difficult to make a hard and fast distinction between malware and bugs. For example, if a programmer left a buffer overflow in a system and it creates a loophole that can be used as a backdoor or a maintenance hook, did he or she do it deliberately? It may not be possible to answer this question with technical means, although we might be able to guess at it, given the relative ease of use of a given vulnerability.

It should be noted that malware is not just a collection of utilities for the attacker, although attack tools may still be legitimate

items of study for software forensics. Once launched, malware can continue an attack without reference to the author or user, and in some cases, will expand the attack to other systems. There is a qualitative difference between malware and the attack tools, kits, or scripts that have to be under an attacker's control, and which are not considered to fall within the definition of malware.

There are a variety of types of malware. Even though functions can be combined, these types do have specific characteristics, and it can be important to keep the distinctions in mind. However, it should be noted that we are increasingly seeing convergence in malware. Viruses and trojans are being used to spread and plant RATs, and RATs are being used to install zombies. In some cases, hoax virus warnings are being used to spread viruses. Virus and trojan payloads may contain logic bombs and data diddlers.

Trojans, or trojan horse programs, are the largest class of malware. However, the term is subject to much confusion, particularly in relation to computer viruses. A trojan is a program that pretends to do one thing while performing another unwanted action. The extent of the "pretense" may vary greatly. Many of the early PC trojans relied merely on the filename and a description on a bulletin board. "Login" trojans, popular among university student mainframe users, mimicked the screen display and the prompts of the normal login program and could, in fact, pass the username and password along to the valid login program at the same time that they stole the user data. Some trojans may contain actual code that does what it is supposed to be doing while performing additional nasty acts that it does not tell you about.

A major component of trojan design is social engineering. Trojan programs are advertised (in some sense) as having some positive utility. The term *positive* can be in some dispute because a great many trojans promise pornography or access to pornography, and this still seems to be depressingly effective. However, other promises can be made as well. A recent email virus, in generating its messages, carried a list of a huge variety of subject lines, promising pornography, humor, virus information, an antivirus program, and information about abuse of the recipient's account. Sometimes the message is simply vague and relies on curiosity.

An additional confusion with viruses involves trojan horse programs that may be spread by email. In years past, a trojan program

had to be posted on an electronic bulletin board system or a file archive site. Because of the static posting, a malicious program would soon be identified and eliminated. More recently, trojan programs have been distributed by mass email campaigns, by posting on Usenet newsgroup discussion groups, or through automated distribution agents (bots) on Internet relay chat (IRC) channels. Because source identification in these communications channels can be easily hidden, trojan programs can be redistributed in a number of disguises, and specific identification of a malicious program has become much more difficult.

Some data security writers consider that a virus is simply a specific example of the class of trojan horse programs. There is some validity to this usage because a virus is an unknown quantity that is hidden and transmitted along with a legitimate disk or program, and any program can be turned into a trojan by infecting it with a virus. However, the term "virus" more properly refers to the added, infectious code rather than the virus/target combination. Therefore, the term "trojan" refers to a deliberately misleading or modified program that does not reproduce itself.

In terms of programming, a trojan probably represents the simplest form of malware. It is, after all, trivial to write a program that will delete a file or format a disk. The only creativity involved relates to finding a cover story that hasn't been used so often that people will get suspicious.

A logic bomb is generally implanted in or coded as part of an application under development or maintenance. Unlike a RAT or trojan, it is difficult to implant a logic bomb after the fact, unless it is during program maintenance.

A trojan or a virus may contain a logic bomb as part of the payload. A logic bomb involves no reproduction and no particular social engineering.

A persistent legend in regard to logic bombs involves what is known as the salami scam. According to the story, this involves siphoning off small amounts of money (in some versions, fractions of a cent) credited to the account of the programmer over a very large number of transactions. Despite the fact that these stories appear in a number of computer security texts, the author has a standing challenge to anyone to come up with a documented case of such a scam. Over a period of eight years, the closest anyone has come is a story about a fast food clerk who diddled the display on

a drive-through window, and collected an extra dime or quarter from most customers.

Also appearing more as a payload is a data diddler. This software deliberately corrupts data, generally by small increments over time. The slow and cumulative damage to the information may not be noticed for some time, and by the time it is remarked, previous backups will probably contain partially contaminated documents.

A hidden software or hardware mechanism that can be triggered to permit system protection mechanisms to be circumvented is known as a backdoor or trap door. The function will generally provide unusually high, or even full, access to the system either without an account or from a normally restricted account. It is activated in some innocent-appearing manner; for example, a key sequence at a terminal. Software developers often introduce backdoors in their code to enable them to re-enter the system and perform certain functions; this is known as a maintenance hook. The backdoor is sometimes left in a fully developed system either by design or accident. Backdoors can also be introduced into software by poor programming practices, such as the infamous buffer overflow error.

DDoS is a modified denial of service (DoS) attack. DoS attacks do not attempt to destroy or corrupt data, but attempt to use up a computing resource to the point where normal work cannot proceed. The structure of a DDoS attack requires a master computer to control the attack, a target of the attack, and a number of computers in the middle that the master computer uses to generate the attack. These computers in between the master and the target are variously called agents or clients, but are usually referred to as running "zombie" programs.

Again, note that DDoS programs are not viral, but checking for zombie software protects not only you and your system, but prevents attacks on others as well. It is, however, still in your best interest to ensure that no zombie programs are active on any of your machines. If your computers are used to launch an assault on some other system, you could be liable for damages.

Most people who actually launch DDoS attacks do not write their own software. Programs to control DDoS networks, slave or zombie programs, and packages to install the zombies are all available on the nets.

The authors of RATs would generally like to refer to these packages as remote administration tools to convey a sense of legitimacy.

All networking software can, in a sense, be considered remote access tools: We have file transfer sites and clients, World Wide Web servers and browsers, and terminal emulation software that allows a microcomputer user to logon to a distant computer and use it as if he or she were on site. The RATs considered to be in the malware camp tend to fall somewhere in the middle of the spectrum. Once a client such as Back Orifice, Netbus, Bionet, or SubSeven is installed on the target computer, the controlling computer is able to obtain information about the target computer. The master computer will be able to download files from, and upload files to, the target. The control computer will also be able to submit commands to the victim, which basically allows the distant operator to do pretty much anything to the prey. One other function is quite important: All of this activity goes on without the owner or operator of the targeted computer getting any alert.

When a RAT program has been run on a computer, it will install itself in such a way as to be active every time the computer is subsequently turned on. Information is sent back to the controlling computer (sometimes via an anonymous channel such as IRC) noting that the system is active. The user of the command computer is now able to explore the target, escalate access to other resources, and install other software, such as DDoS zombies, if so desired.

Once more, it should be noted that remote access tools are not viral. When the software is active, though, the master computer can submit commands to have the installation program sent on, via network transfer or email, to other machines.

Rootkits, containing software that can subvert or replace normal operating system software, have been around for some time. RATs differ from rootkits in that a working account must be either subverted or created on the target computer in order to use a rootkit. RATs, once installed by a virus or trojan, do not require access to an account. Once again, rootkits themselves may not be considered malware, although they can certainly be used for malicious purposes.

Other programs in this gray area between utilities and malware are sniffers. Sniffer packages essentially provide for eavesdropping on computer networks. Although they do not allow an avenue to information on machines, they do provide access to any data that flows between devices. This information can, of course, involve information about the structure and protections of the systems, including passwords and similar entry codes.

Pranks are very much a part of the computer culture. So much so that you can now buy commercially produced joke packages that allow you to perform "Stupid Mac (or PC, or Windows) Tricks." There are numberless pranks available as shareware. Some make the computer appear to insult the user; some use sound effects or voices; some use special visual effects. A fairly common thread running through most pranks is that the computer is, in some way, nonfunctional. Many pretend to have detected some kind of fault in the computer (and some pretend to rectify such faults, possibly making things worse). One entry in the virus field is PARASCAN, the paranoid scanner. It pretends to find large numbers of infected files, although it doesn't actually check for any infections.

Generally speaking, pranks that create some kind of announcement are not viral, and viruses that generate a screen or audio display are rare. The distinction between jokes and trojans is harder to make, but pranks are intended for amusement. Joke programs may, of course, result in a denial of service if people find the prank message frightening.

One specific type of joke is the "Easter egg," a function hidden in a program, and generally accessible only by some arcane sequence of commands. These may be seen as harmless, but note that they do consume resources, even if only disk space, and also make the task of ensuring program integrity very much more difficult. The presence of an Easter egg will definitely have an impact on software forensics, on the one hand increasing the volume of material that must be assessed, and on the other hand providing a potentially larger sample for comparison or statistical purposes.

Other Products

As noted, the activities of the blackhat community are primarily of interest when we are considering malicious software of various types. Software forensics may be used in a number of other cases, particularly in regard to intellectual property. In the case of stolen software, financial reward may be the sole consideration, and a random opportunity the only means. The perpetrator may not have any characteristics in common with the blackhat community.

In some situations, the theft of software may have a relation to the "warez" group noted earlier. However, it is unlikely that these anarchic individuals could put together a company selling commer-

cial software, particularly the mass market variety. The likelihood is somewhat greater in relation to specialized niche software because corporate structure, distribution systems, and mass marketing has less of a role to play.

There is, of course, always the possibility of the theft of a certain piece of commercial code for inclusion in, say, an open source software project. This type of activity is more likely to be motivated by the same kind of ego drive we have noted is very important in blackhat circles. However, open source devotees are likely to spot this type of theft themselves, and will almost certainly reject any such donations in favor of home-grown versions.

Summary

A significant fraction of the work that software forensics may be called on to examine will have to do with malicious software, and other products of the blackhat communities. Therefore, without blinding ourselves to other possibilities, it is good to have a rough idea of the stereotypical characteristics we might expect to find in the authors of these programs. The attributes may be apparent in various aspects of the programming we are called on to examine.

The blackhat communities may be united in aspect, but they are very much divided in aims, activities, and nominal skills. Having an understanding of the various types of groups can be useful in regard to identifying the cultural influences that we observe when dealing with a specific piece of software.

It is important to understand that blackhat motivations may be substantially different from those found in other types of illicit activity. For those who are used to dealing with primarily profit-driven criminal behavior, the predominately ego-driven blackhat performance may be difficult to comprehend.

The various types of blackhat-produced software will come from different groups, with differing aims and skill sets. Therefore, it will be useful to have a realistic grasp of the multiplicity of types of software and the objectives behind them.

Having outlined this one broad class of objects of study, and the people behind it, we now turn to the more general description of the production of software code and the tools we will be using to examine it.