Malicious Cryptography

Exposing Cryptovirology

Adam Young

Moti Yung

WILEY

Wiley Publishing, Inc.

Malicious Cryptography

Malicious Cryptography

Exposing Cryptovirology

Adam Young

Moti Yung

WILEY

Wiley Publishing, Inc.

Executive Publisher: Robert Ipsen
Executive Editor: Carol A. Long
Developmental Editor: Eileen Bien Calabro
Editorial Manager: Kathryn A. Malm
Production Manager: Fred Bernardi

This book is printed on acid-free paper.

Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

ISBN: 0-7645-4975-8

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Dedicated to Elisa (A. Y.)
and to Maya (M. Y.)

Contents

Foreword

Terms such as cryptovirology, malware, kleptogram, or kleptography may be unfamiliar to the reader, but the basic concepts associated with them certainly are familiar. Everyone knows—often from sad experience—about viruses, Trojan horses, and worms and many have had a password "harvested" by a piece of software planted surreptitiously on their computer while browsing the Net. The realization that a public key could be placed in a virus so that part of its payload would be to perform a one-way operation on the host computer that could only be undone using the private key held by the virus' author was the discovery from which *Malicious Cryptography* sprang. Rather than describe these notions here, intriguing as they are, I'll only try to set the stage for the authors' lucid description of these and other related notions.

Superficially, information security, or information integrity, doesn't appear to be much different from other functions concerned with preserving the quality of information while in storage or during transmission. Error detecting and correcting codes, for example, are intended to ensure that the information that a receiver receives is the same as that sent by the transmitter. Authentication codes, or authentication in general, are also intended to ensure that information can neither be modified nor substituted without detection, thus allowing a receiver to be confident that what he receives is what was sent and that it came from the purported transmitter. These sound remarkably alike in function, but they are fundamentally different in ways that are at the heart of *Malicious Cryptography*. The greatest service this Foreword can render is to give the reader a crisp, clear understanding of the nature of this difference in order to set the stage for the book that follows.

Most system functions can be quantitatively specified and tested to verify that the specifications are met. If a piece of electronic equipment is supposed to operate within a specified range of a parameter (such as voltage, acceleration, temperature, shock, vibration, and so forth), then it is a straightforward matter to devise tests to verify that it does. Closer in spirit to information security and integrity than physical environmental specifications would be a specification of a communication system's immunity to noise or bit errors. One might specify the minimum data bandwidth for a given signal to noise (SN) ratio or the allowable bit error rate. Again it is a straightforward matter to devise tests that verify the data bandwidth or the bit error rate for a signal possessing the specified signal to noise ratio. Error detecting and correcting codes may be tailored to the expected statistical nature of the noise, Fire codes for burst errors or Grey codes for an angular position reading device, etc. But the verification that the system is meeting specifications remains straightforward and quantitative.

Security is fundamentally different from any other system parameter, however. One of the largest alarm and vault manufacturers in the U.S. discovered this in a costly example a few years ago. Vaults and safes are routinely certified for the time documents will survive undamaged in a fire—itself specified by temperature and type (oil, structural, electrical, etc.). They had developed a new composite material that was very resistant to cutting, drilling, burning, etc. Extensive tests had been conducted with cutting tools of all sorts including oxyhydrogen burning bars, drilling with mechanical drills and hypervelocity air-abrasive drills, etc. Based on these results, they guaranteed their safes and vaults made of the new material would provide a specified minimum time for penetration. What they had overlooked was that linear cutting charges (shaped charges) that were widely used in the oil industry for cutting oil well casings and in the demolition business for slicing building supports to bring down buildings could be used to cut out a panel from the side of a safe or vault in milliseconds instead of requiring hours. This long aside is very germane to this Foreword. The safe and vault company had measured the resistance of their product to the attacks they anticipated would be used against them. The robbers used an entirely unexpected means to open the vault—and the company paid dearly for their oversight. *Malicious Cryptography* is almost entirely about doing things in completely unexpected ways in information integrity protocols.

Going back to the example with which we started, the fundamental difference between error detecting and correcting codes and authentication, both of which function to ensure the integrity of information, is that the first is pitted against nature and the other against a human adversary. Nature may be hostile, the signal to noise ratio may be large, the signal may drop out for extended periods of time, other signals may randomly mask the desired signal, but nature is neither intelligent nor adaptive. A human opponent is both. He may also be interactive, probing to gain information to allow him to refine and adapt his attacks. As those of us in the information security business like to say, there is no standard attacker and no standard attack. This is in contrast with all other specifications where standard environments, no matter how hostile or unpredictable, are the norm.

What the authors of *Malicious Cryptography* have done very successfully is to capture the essence of how security can be subverted in this non-standard environment. On several occasions, they refer to game theory without actually invoking the formalism of game theory—emphasizing instead the game-like setting in which security is the value of the ongoing competition between a system designer and its attackers.

There have been many books on hacking, software subversion, network security, etc., which consist mainly of descriptions of successful attacks—some exceedingly clever and many very devious in their execution. These are similar in style and feeling to Modern Chess Openings (MCO) that every chess player knows, studies, and on which he depends. There are of course many possible lines of play in chess, but the several hundred openings that have stood the test of time and repeated tournament play make up the MCO. Roughly the first twenty moves or so of these openings, with promising variations, have been so thoroughly analyzed and understood that it is rare indeed for an opening not in the MCO to be successful in match play. A similar situation is true for the end game—not that the endings are so cataloged and restricted, but rather that the game has simplified to where almost a counting-like analysis reveals the outcome to a knowledgeable player. Masters will resign a game as lost at a point where a less experienced player may not even be able to see who has the advantage. As most books on hacking recount one clever attack after another, MCO recounts one opening after another with an ! or !! in the annotation to flag a particularly brilliant move. I almost expect to find an exclamation mark in the margin of most books on software subversion when the deception on which a particular protocol failure turns is revealed.

The middle game in chess, though, must be guided by general principles since the number of lines of play—the attack, counter attack—between two masters is virtually unlimited. So it is with information security protocols and cryptosystems. The possibilities are virtually unlimited so general principles must guide both the system designer and the counter designer; the attacker seeking to exploit hidden weaknesses in the design; the designer seeking to prevent such attacks or failing that, to detect them when they occur. *Malicious Cryptography* pioneers in motivating and clearly enunciating some of these principles.

Cryptography, authentication, digital signatures, and indeed, virtually every digital information security function depend for their security on pieces of information known only to a select company of authorized insiders and unknown to outsiders. Following the usual convention in cryptography we will refer to this privileged information as the key although in many situations the only thing in common with the usual notion of a cryptographic key is that it is secret from all but a designated select few. It may well be that no individual knows the key but that a specified set of them have the joint capability to either recover it (shared secret schemes) or to jointly execute a function that in all probability no outsider or any proper subset of them can do (shared capability schemes). It is almost always the case that this secret piece of information is supposed to be chosen randomly—from a specified range of values and with a specified probability distribution, generally the uniform distribution. The assumption is that this insures that an unauthorized user will have no better chance of discerning the secret key than the probability the same key will be drawn in an independent drawing of a new value under the same conditions. It is also generally assumed that only the person choosing the random number knows it. In fact he may share it with someone else at the time it is drawn, or they may have chosen the number in advance of the supposed drawing. In the most extreme case it may be dictated by some other participant and not chosen by the person supposed to be choosing it all. Every one of these surreptitious variants has been the basis for serious subversions of information integrity and security protocols. One of the central themes in *Malicious Cryptography* is the mischief that is possible if these conditions are not met; in other words, if the "random" value is not random in the sense supposed.

Since security or integrity is directly measured by the probability the secret key can be discovered (computed) by unauthorized cabals of attackers, the information content of the key (roughly speaking, the size of the random number) must be great enough that it is computationally infeasible to simply try all possible values—known as a brute force key space search. But this means that it is then computationally infeasible for a monitor to tell whether the random values produced were actually randomly chosen as supposed or not. This is at the heart of subliminal channels, for example. The subliminal transmitter and receiver share in secret information about the bias imposed on the selection of the session keys which enables them to communicate covertly in the overt communications while it remains computationally infeasible (impossible?) for a monitor to detect a bias in the session key selection process, and hence impossible for him to detect either the presence or use of a subliminal channel.

The dilemma is that if the key is large enough to be secure, it is also large enough to make it impossible to detect a bias in the selection process. It therefore becomes possible to hide information in the keys, to communicate other keys subliminally, to make it computationally feasible for designated receivers to perform a key space search while a full search remains computationally infeasible for outsiders to do, to subvert information integrity protocols from within, etc. The list of possible deceptions is virtually unlimited and the authors of *Malicious Cryptography* have exploited many of these in innovative ways.

In information integrity protocols nothing can be taken for granted, i.e., nothing can be assumed that cannot be enforced. If the protocol calls for a number to be chosen from a specified range using a particular probability distribution, then the assumption must be that it isn't unless the other parties to the protocol can force it to be in a secondary protocol. Otherwise you must assume it could be chosen from a restricted range or chosen using a different probability distribution, or that it was chosen earlier and shared with persons assumed not to know it, or that it isn't being selected at random at all by the person supposed to be choosing it, or that it is dictated to him by another party not even considered in the protocol. Several of the subversions described in *Malicious Cryptography* depend on this ability to undetectably hide information in keys. The point germane to this Foreword, though, is that it is the general principle that is vital for both the designer and the counter-designer to keep in mind. There are interactive protocols to insure that the objectives of randomness are met. Those protocols are not the subject of *Malicious Cryptography*, but made all the more important because of the weaknesses exposed in it.

There are other examples, though, in which no means is known to enforce the desired outcome. Several protocols call for a public modulus to be the product of two secret primes chosen so as to make it computationally infeasible to factor the modulus—usually only a function of the size of the factors although in some protocols the factors must satisfy some number theoretic side condition such as belonging to a particular residue class, etc. It is possible to work a variety of mischiefs if a modulus that is the product of more than two prime factors can be passed off as the product of only two. In particular, a subliminal channel becomes possible with the desirable feature that while the subliminal receiver can receive subliminal messages sent by the transmitter he cannot falsely attribute a forged message to the transmitter. It is only polynomially difficult to distinguish between primes and composite numbers. But so far as is known it is just as hard to tell if a composite number has three or more factors as it is to factor the number itself! In the absence of an interactive protocol to ensure that a modulus has two and only two prime factors, deceptions that depend on the existence of three or more factors remain a possibility. Deceptions of this sort do not appear in *Malicious Cryptography* and are mentioned here only to illustrate that not all general principles for deception have solutions available to the designer at the moment.

Malicious Cryptography is a remarkable book; remarkable for what it attempts and remarkable for what it achieves. The realization that cryptography can be exploited to achieve malicious ends as easily as it can to achieve beneficial ones is a novel and valuable insight—to both designers and counter-designers of information security and integrity protocols.

Gus Simmons

September, 2003

Acknowledgments

We have so many people to thank that it is difficult to figure out where to begin. It has been said that ideas cannot be created in a vacuum and in this we believe wholeheartedly. *Malicious Cryptography* is the product of interactions and collaborations that span over a decade. In truth we have family, friends, teachers, coworkers, researchers, students, anonymous referees, journalists,[1] science-fiction authors, movie writers, artists, and musicians[2] to acknowledge. Without such support, enthusiasm, artistic creativity, teachers, and listeners, this book would not have been possible.

First and foremost we thank Columbia University, our mutual alma mater. It was at Columbia that our research began, and it was at Columbia where we met a great number of brilliant people from whom we learned, and with whom we worked and shared ideas. We thank Zvi Galil, Dean of the School of Engineering and Applied Science, who served as faculty advisor to us both. We thank Jonathan Gross and Andrew Kosoresow, both of whom served on Adam's PhD committee. Andrew was a great and dedicated educator, and we mourn his untimely passing. We thank Matt Franklin and Stuart Haber, both of whom graduated from Columbia. Matt and Stuart have served as collaborators to us both as well as lecturers in graduate courses taken by Adam. On numerous occasions Adam flew into Matt Franklin's office, wide-eyed and somewhat insane looking, for the sole purpose of scrawling a brand new attack on his blackboard just to see how he would react. Adam also thanks Matt Blaze for teaching an inspiring course on computer security in 1995 and for fostering great interest in cryptography among his students. Moti extends his gratitude to

[1] John Markoff, Steven Levy, Katie Hafner, and Bruce Sterling among others.

[2] Adam thanks Nine-Inch-Nails, Sonic Mayhem, White Zombie, Looking Glass Studios (System Shock 2 Soundtrack), Devo, and Danzig for setting the mood for the beginning of the book.

all of his coauthors and everyone he has worked with over the years, since it is through scientific work and the exchange of ideas that one develops as a researcher.

We thank Markus Jakobsson from RSA Data Security. Moti mentored Markus throughout his dissertation defense preparation and Markus in turn served on Adam's PhD committee. Markus reviewed this text and has sponsored annual lectures on Cryptovirology at NYU. We thank Yiannis Tsiounis, another student that Moti assisted, for sharing ideas and for reviewing this book. We thank our colleague Yair Frankel for sponsoring an invited lecture on kleptography for the Information Surety Group at Sandia National Labs. We thank Michael Reiter for supporting Adam while at Lucent Technologies in the Secure Systems Research Division, and for hosting a lecture on subliminal channels and kleptography.

Adam thanks Matthew Hastings from Los Alamos National Laboratory. Over the course of four years at Yale, Matt and Adam jointly experimented with self-replicating code in a safe and controlled environment. Many of the discoveries and open problems that were found gave impetus to investigating advanced malicious software attacks. Adam also thanks Mark Reed from the Yale University Department of Electrical Engineering. Mark served as Adam's undergraduate faculty advisor and provided support for his career both inside and outside of the classroom.

Adam thanks Cigital Labs and in particular Jeff Voas, Jeff Payne, Gary McGraw, and Matt Schmid for encouraging this work. We thank Christoph C. Michael, senior research scientist at Cigital Labs, for engaging conversations, contributing artwork, and for lending an ear to a never-ending stream of clandestine malware rhetoric. We also thank Alexander Antonov and Paul DesRivières from the Cigital Secure Software Group for reviewing the manuscript line by line and Mike Copenhafer, Bruce Potter, Mike Firetti, Viren Shah, Frank Hill, Coleman Baker, and Chris Ren from Cigital for helpful reviews and discussions.

From Wiley we thank Carol Long,[3] Eileen Calabro, Fred Bernardi, Robert Ipsen, and Kathryn Malm. Carol and her team produced this book in remarkably short order with the utmost degree of professionalism.

Special thanks goes to Dmitriy Pozdnyakov, Michael Makarius, Leo C. Petroski, and H. Robert Feinberg for helpful feedback and overall support of this work. Finally Adam would like to thank his wife, Elisa Young, for being. Without her this book would cease to have meaning.

[3]Or *cryptolady*, as she is known at Wiley.

Introduction

This book is a compendium of malicious software and hardware attacks geared towards subverting computer systems. The attacks are not of the sort that exploit software bugs, design flaws, and so forth. The business of bypassing security measures is outside the scope of this work. Rather, we present a series of cryptographic methods for defiling computer systems once internal access is acquired.

Some of the attacks are more technical than others, involving recent advances in the field of cryptology. As a result this book is likely to be received in a variety of different ways. To hackers it may serve as a vade mecum. To security professionals it may serve as a long overdue warning. To science fiction buffs it may serve as a good read, and to intelligence agencies it may serve as a challenge to our First Amendment rights.

Chapter 1 is a motivational chapter that portrays the world through the eyes of a hacker. It reveals the very fabric of a hacker's existence and due to its illicit nature we mention the standard disclaimer that reads, "do not try this at home." To perform any of the acts described therein is to risk violating the Computer Fraud and Abuse Act of 1986, among others. Hackers face *scientific* problems when trying to infiltrate computer systems. It was by experiencing these problems first hand that many of these attacks were discovered.

A great number of people share a close kinship with our digital brethren and to hackers it is no different. But whereas to writers it is through text, to artists it is through images, and to musicians it is through music, to hackers it is through the very language that computers speak when speaking with each other, the language of *binary*. To speak in binary and hear every word they say is to be one with the machine and that feeling can be hopelessly and utterly addictive.

To the uncorrupt of spirit the need to join with the machine can be controlled to a degree. This need is illustrated in Chapter 1 over the course of three short stories. They are written in second person singular and as such force the reader to play the role of the subduer. It is the reader that steals passwords using a Trojan horse program. It is the reader that spends years developing an insidious computer virus, and it is the reader that takes over the local area network of a small company. Yet everywhere in the storyline the privacy and integrity of other people's data is respected. It portrays the pursuit of knowledge and the thrill of the hunt, not the kill.

As Lord Acton once said, "power corrupts; absolute power corrupts absolutely." This could not be truer with respect to hacking. For this reason we urge readers not to abuse the ideas presented in this book. If our efforts coax so much as a single hacker to embrace the greater mathematical challenges facing system security, then our writing will not have been for naught, for such a hacker is likely to seek recognition in the form of conference papers in lieu of news reports.

Given the clandestine nature of the algorithms and protocols that are presented, it is important to emphasize the nature of secure systems research. Cryptanalysis exists to help make cryptosystems more secure. *The goal of cryptanalysis is not to undo the honorable work of others, but to find vulnerabilities and fix them.* Many a cryptographer has suffered the disheartening realization that his or her cipher has been broken. Lucky are those who discover this themselves, but many are they who learn the hard way when another researcher publishes the discovery in an academic forum. Cryptanalysis is the mathematician's version of hacking: it is both devil's advocate and antithesis of cryptography. History has proven the need for cryptanalysis and hence the need to find weaknesses in cryptosystems and publish them. It may be reasoned that the need for cryptanalysis extends directly to the need to investigate attacks on modern computer systems. This, we argue, is the realm of cryptovirology and in this treatise we take a first step in this direction.

In the public eye, the word *cryptography* is virtually synonymous with *security*. It is a means to an end, a way to send e-mail privately and purchase items securely on-line. If nothing else this book will challenge that view. In the chapters that follow it is shown how modern cryptographic paradigms and tools including semantic security, reduction arguments, polynomial indistinguishability, random oracles, one-way functions, Feistel ciphers, entropy extractors, pseudorandom number generators, etc., can in fact be used to *degrade system security*.

It is shown how to devise a cryptovirus to usurp data from a host machine without revealing that which is sought, even if the virus is observed at every turn. It is shown how to design a password-snatching cryptotrojan that makes it virtually impossible to identify the author when the encrypted passwords are retrieved. Furthermore, it is intractable to determine if the cryptotrojan is encrypting anything at all even when it is under constant surveillance.

Still other cryptotrojans are described that attack industry-standard cryptosystems. By design, these Trojans give the attacker covert access to the private keys of users and are extremely robust against reverse engineering. When implemented in tamper-resistant devices these transgressions cannot be detected by anyone save the attacker. Such Trojans are ideal for governments that wish to obtain covert access to the encrypted communications of their citizens. These Trojans show how to apply cryptography within cryptography itself to undermine the very trust that cryptosystems were designed to provide. In so doing we will expose the dark side of cryptography and thereby reveal its true dual-edged nature.

Several of the attacks have known countermeasures, some of which are ideal and others that are merely heuristic in nature. These defenses are described in detail to give the book a more balanced presentation to the community at large. It is our belief that these malicious software attacks should be exposed so that security analysts will recognize them in the event that they appear in fielded computer systems. Doing so has the potential of minimizing the malicious software learning curve that practitioners might otherwise face.

In all likelihood the attacks that are described in this book constitute the tip of the iceberg in terms of what is possible. Offensive information warfare is an area of research that is scarcely funded by the U.S. government, for obvious reasons. However, the notion of malicious software as well as cryptography is by no means new to the federal government, and so one would expect that there has been more classified research in this area than unclassified research. This book is our earnest attempt to expose the open research in this area, since corporations, governments, and individuals have a right to know about that which threatens the integrity of their computing machinery.

Some readers will inevitably object to the nature of this book. To this end we remark that these attacks exist, they are real, and that it is perilous to sweep them under the rug. We believe that they will surface sooner or later. It is our hope that this book will encourage the study of cryptography as a whole and at the same time reveal some of the more serious threats that computer systems face, both *from within* and from without.

A. Y.

M. Y.

October, 2003

Chapter 1

Through Hacker's Eyes

There is no way to describe the feeling of approaching a computer system to download the data that your Trojan horse has been collecting for days. Your heart begins to race. You look over your shoulder out of instinct and start to have major second thoughts about proceeding. The computer terminal is unoccupied and sits directly in front of you.

Questions plague your thoughts: How many people are capable of finding the cleverly hidden Trojan? More importantly, does anyone *in this room* know it is there? You ease yourself down into the chair. Glancing to your right you see a student stare at his calculator with a perplexed look on his face. To your left a girl is laughing on her cell phone. If you could shrink yourself into nothing and crawl through the cracks in the machine you would gladly do so. But you are physical and there is nothing you can do about that now. The coast is clear. You reach for your floppy and insert it into the drive. Sheens of sweat glaze over your palms. Why? Because after all, you are returning to the scene of a crime.

Your crime.

Deep down, you rationalize your actions. There is no blood involved, no money is being stolen, and in the end no real harm is being done. . . or is there? The floppy drive begins to spin. In moments it will be over. In moments all of the login/password pairs will be on the disk and you will be hightailing it to your next class. Perspiration breaks out on your forehead but is easily dismissed with a waft of your hand. You navigate to the floppy drive and double-click on the game of Tetris. There is time for one quick game. The first block is 1 by 5, your favorite. If only they'd come down like that one after the other you'd have the game in the bag by laying them out horizontally. But it never works out that way. The law of probabilities won't allow it. A book hits the ground and you jump.

A lanky-looking freshman picks it up. The title—*Differential Equations: Theory and Applications*. Smart guy. Most students are only studying multivariable calculus in their first year. Words begin to echo in the back of your mind: *there has to be a better way, there has to be a better way. . . .* An odd, misshapen block comes into view. You hate those. They make you lose Tetris every time. A whirring noise emanates from the drive and this time you know it is writing to the floppy. One more minute and your doctored up version of Tetris will have downloaded all of the passwords to the disk. Who'd ever guess this version of Tetris packed such a punch?

A four-sided cube comes down and you ease it over to the left-hand side of the screen. You love those shapes too. On the surface you are just playing a game. Your mouse button clicks and space bar presses are as innocuous as they come. But the real game you are playing is not so easy to see, and at times it feels like Russian roulette. Your thoughts wander to your password-snatching Trojan. The possibility that it was found and that silent sysadmin alarms are sounding *in a nearby room* is very, very real.

Something's wrong.

Something's not right; you can feel it in the air. The drive should have stopped spinning by now. Your heart goes still. Looking up, you catch a glimpse of a man you didn't notice before. He makes eye contact with you. Fighting the urge to flee, you quickly look back at your screen. You missed placing two blocks. You will not make high score. Your mind begins conjuring swear words without biblical precedent. . . it has *never* taken this long before.

The floppy drive finally stops whirring. You quit out of Tetris, eject the floppy, and reboot the machine. You leave the computer cluster and enter the hallway half expecting to be halted by university officials. But none are there. You think yourself silly. You think that there was no way it could have been found. But the reality is that you know all too well how to write a background process capable of catching you in the act and that is what makes you scared. Stepping outside the building, you breathe a sigh of relief in the midday sun. You made it this time, but maybe you were just lucky. Maybe it wasn't in the cards just yet. Like a junkie to drugs, you are drawn to these machines. They speak to you the way they speak to no one else. You put in your time. You paid your dues, and yet for some reason your vision is still shrouded in darkness. There is something they are not telling you. Perhaps it is something they don't even know. It is a question that nags at you like no other, and you sense that the answer lies hidden somewhere within the deepest recesses of your soul, somewhere out of sight and just beyond your grasp. *There has to be a better way.*

Shortly before sundown that same day...

The dull roar of thunder reverberates somewhere far off in the distance as menacing storm clouds roil in from the west. They exhibit all the signs of a true nor'easter and threaten to engulf the entire city of New Haven. You swear you just felt a drop of rain hit your left shoulder. Reaching down, you feel for the disk at your side. The floppy is still there, its presence reassured at the touch of a thumb. The data it contains is dear to you, and you'll be damned if you're gonna let a little H_2O seep through your denim pocket and claim your catch of the day. So you decide to pick up the pace a bit.

The path you follow winds in and around, gently sloping downward as you go, eventually leading to a clearing that overlooks a stand of maples. The trees are enormous and have stood here for ages. At their center lies a lone apple tree. It is dwarfed by the older trees and is helplessly sheltered under a canopy of leaves. Having sensed your unexpected approach, a nearby squirrel dashes for the safety of a nearby tree. Before reaching the trunk, it fumbles over an apple and sends it rolling along the ground. The fruits around you give off a racy odor, a telltale reminder of the approaching change of season.

Had it not been for the disk, you would chance a brief pause underneath the eaves to contemplate greater things. Physics lectures always left you spellbound regarding the mysteries of the world. It was the dream of being struck in the head by a falling apple that guided you to this school in the first place, a dream that you summarily dismissed upon meeting your brilliant roommate. He is a National Merit scholar and received 1580 out of a possible 1600 on his SATs. The deduction was in the verbal section, and you always attributed it to his difficulty in comprehending the human condition. On many levels he is more machine than man, yet his inference engine is second to none. Physics is his second language and he speaks it fluently. You abandoned the idea of majoring in physics since the thought of taking the same classes as he was too much to bear, and since he had an uncanny ability to make you feel stupid without even trying. Answers to scientific problems just came *naturally* to him. Your hacking obsession combined with a thoroughly tenderized ego would do little to help you finish school.

A gust of wind billows through the trees. The limbs creak and sway in response, causing rain droplets to roll off their leaves. The water splashes onto your face and exposed arms, causing you to start. You realize that you had zoned out completely and had lost all track of time. Your eyes had stared off into space, fixated on some solitary trees, and subconsciously absorbed the surrounding scenery. You shrug in spite of yourself. No use in crying over spilled milk. Your true path has yet to be determined and there is no reason to worry about it now.

You shift the weight of your backpack to your other shoulder and leave the small wooded area behind. As always the students took Prospect Street back to Old Campus while you ventured along an overgrown yet shorter route, preferring to take the road less traveled. *Hypotenuse* action your roommate called it. Over time you discerned the shortest route between the Sloane Physics Lab and your dorm and it took you through more than one private yard, not to mention a vast cemetery. It saved you an innumerable number of backaches to be sure. Take aside any science student and you will hear the same tale of woe. The cumbersome textbooks are murderous to haul and the university couldn't place the science buildings at a more remote location if it tried.

The Payne Whitney Gymnasium looms ahead, shadowed by the black storm cover above. Were it not for the parked cars and street signs, the darkness could easily lead one to mistake it for a castle. Gulls from the nearby seashore circle above the parapets that line the rooftop. Some dive and soar, some pick up speed, and still others hover in place in blatant defiance of the wind. Nightfall descended prematurely on the city, and what had been just a few droplets of rain minutes before has turned into a veritable deluge. A small pack of students run through the stone archway at the base of the gym with newspapers outstretched overhead. The brunt of the storm is upon you and rainwater quickly seeps into every quarter. You break into a sprint down Tower Parkway in a last-ditch effort to keep your data dry.

The torrential rain pummels your body in sheets as you approach the backdoor of Morse College. You pass quietly into the building under cover of dusk and enter the underground labyrinth of steam tunnels and storage rooms. The humdrum of washers and dryers from a nearby laundry room fills your ears. You take a brief moment to wring what water you can from your clothing. After regaining your composure, you head down the narrow hallway and pass alongside the laundry room. It is empty and devoid of movement, save for a loose ball of lint circling beneath a ventilation shaft.

You continue along the corridor towards the small staircase at its end, leaving a puddle of water with each passing step. A steam release valve hisses as you pass it by, only to be replaced by the distant clamor of trays and dishes. The student body has assembled in the Morse cafeteria for the high-quality food service afforded by the university. It is the early part of dinner hour and the thought of eating couldn't be further from your mind.

You fish the keys out of your pocket as you gain the steps to your floor. If your roommate is in he'll probably give you a hard time about tracking water inside, and rightfully so. You open the door and swing it wide, revealing the darkened room beyond. He's out, probably studying in the science library as usual. You pass through his room and into yours, opting to leave the lights out for fear of ruining the picturesque atmosphere. With the toil of the long trek behind, you ease your backpack to the ground and rest at the foot of your bed. You suspect that he'll be gone for the better part of the evening.

It is nights like these that you live for.

A momentary flash of lightning illuminates every darkened corner of the room. You are not alone. A woman stares at you from across your bed. Her eyes are as cold as ice and she has daggers at her sides, drawn at the ready. *Li* could lunge at you at any moment. It is perhaps one of H. R. Giger's most beautiful yet grotesque works of art ever, and you purchased the poster for twenty dollars at The Forbidden Planet in Manhattan.[1]

Is she man or machine? Does she need blood or electricity to survive? Perhaps she needs a bit of both. No one really knows of course, no one except H. R. Giger himself. But the purpose of the metal sheaths is clear. They were carefully designed to extract every last drop of blood for her consumption. Her face is paradoxical: it is clearly frozen in a state of suspended animation, yet her eyes are seeing and behind them she is actively calculating. Li has all the makings of perfection: the memory capacity and precision of a supercomputer, the ability to reason as humans do and perform *modus ponens*, yet exist free of fear and pain and want, with the life expectancy of a *machine*. There is a definite eeriness about her, for her eyelids are at half mast and she gives off the impression of total boredom, as if it is out of curiosity alone that she permits you to gaze upon her before taking your life.

After a time you get up and seat yourself at your computer, feeling

[1]The mesmerizing 56" × 80" original is entitled "Li II" and hangs in the Swiss Art Museum (see http://www.giger.com).

her eyes penetrate deep into the back of your head as you do so. She has watched all of your feeble attempts at becoming one with the machine.

The disk is soaking wet. You pull it out and lay it down next to your keyboard. The writing on the label is smeared beyond recognition. A blow dryer simply will not do, and neither will a tissue since it can leave nasty scratches if sand gets in the way. It will require surgery to salvage it on such short order. You remove the sliding metal door causing a small metal spring to fly out and fall to the ground. The door is warped irreparably, but you will not be needing it again. The two plastic halves separate easily and you gingerly extract the silicon disk from its casing. It has water droplets all over it. They are not too big, but it's a good thing you didn't insert the disk into your drive. You take a dry towel from the bathroom and lay it out on the desk, carefully placing the thin silicon platter on top of it. The water will evaporate soon enough.

You draw your attention to your computer. The power is still off and the pen that you positioned carefully atop the keyboard has not moved. The upper end rests squarely between the "5" and "6" keys and the ball-point end lies between the "c" and "v" keys. Had it not been aligned as such there would have been hell to pay, and the inquisition would have commenced with your roommate. You remove the pen and flip on the power switch. The desktop appears. The background art reads "Night City" haphazardly spray painted along a worn and weathered wall set beneath a neon sky. The steel rods from the reinforced concrete stand rusted and jagged along the top, making for rough passage should anyone try to reach the ruined building beyond. You dubbed the machine Night City in honor of the cyberpunk role-playing game that bears the same name.[2]

The protagonists in the cyberpunk genre are a truly admirable lot. They are high-tech lowlifes that challenge authority at every given opportunity, blend in with the crowd, and make commercial programmers look like toddlers playing with tinker toys. The *sprawl* is their home, a megalopolis formed from the eventual unification of Boston, New York, and Philadelphia. The cyberpunks live on the fringes of society and form a counterculture unto themselves. They know not of greed. They know not of rapacity, and they know not of hegemony. However, these things are not alien to them since they are contended with on a regular basis. They are technologists absolute and embrace mankind's tendency to both make its own problems and later overcome them: deplete the ozone then sell sun block; pollute the air then sell gas masks; trash this planet then

[2] See [27, 229].

move on to the next. It is in science and technology that they *believe*. Like renegade cowboys out of the Wild West, they serve their own needs in the largely lawless and uncontrollable digital realm. Yet they frequently perform valuable services for the common good, and play a crucial role in keeping the powers that be in check, thus preserving the freedoms that we take for granted. In the end their heroic acts are seldom if ever rewarded, let alone recognized. Such is the divine tragedy of the good hacker. *When the megacorporations of the world and their puppet governments wrest control of our lives completely, when they see and hear and record every move we make, when they tell us how we should think and how we should act and what we should buy, who else will there be to turn to?*

The terminate-and-stay-resident programs load one after another, creating a line of icons along the bottom of the screen. After the last one loads you reach around the left side of the machine and press the hardware debugging switch. Time to go manual. You type in a command to view the two bytes located at address 0x05DE1940. It contains 0x007E, just as it should. It read 0x007D when you left, implying that you are the only one who booted the machine since you went to class this morning. Your computer is running a number of custom-made Trojan horses, and this is the result of one of them. Every time the machine boots the Trojan increments the counter by one.

On one occasion you rebooted the machine and found that the value had been incremented by two. After a prompt interrogation of your roommate you learned that he had turned on your machine to see if you had some software he needed. When he was finished he turned the machine back off. Paranoia perhaps? Well, call it what you will. You regard it as a simple matter of dotting your i's and crossing your t's. Anyone who walks more than 10 feet inside *Night City* will set off one alarm or another. There are those who would search your machine, if not for your list of pilfered passwords, then for evidence regarding your other extracurricular activities. *Trojans help solve this problem too.* Any such person would only find ciphertexts and a machine so riddled with custom-made Trojans as to lead one to wonder why you hadn't written the operating system from the ground up in the first place.

There was no need to admonish your roommate for using your computer. You trusted him more than anyone else in the world with its contents. He had won your respect on the first day of school due to his raw intellect alone. There seemed to be no question he could not answer, no system of equations he could not solve. This applied to everything, from using Maxwell's equations to describe an electrical phenomena to figuring out how computer viruses worked. This had its downsides of course, since there is nothing more frustrating than knowing that whenever you got stuck on a homework problem, the oracle in the adjacent room could produce the answer in a matter of seconds.

The stage has been set. Soon the disk will be dry and you will be able to read its contents. You lean back in your chair and throw your hands behind your head. Ruminations of the previous lecture take over your thoughts. It was a class on the history of physics, and was taught by Professor Klein. He is one of the world's foremost authorities on the subject, and what adds greatly to his lectures is the fact that he even looks like Albert Einstein, although you'd be hard-pressed to get another classmate to admit it openly.

His lecture centered on Neils Bohr, the 1922 winner of the Nobel Prize in physics. It was awarded for his successful investigations on the structure of atoms and the radiation emanating from them. However, as Professor Klein explained, his contributions to mankind far exceeded his status as a Nobel laureate. He was arguably deserving of a peace prize as well for his heroic efforts at saving Jews from Nazi tyranny. Under threat of complete Nazi dictatorship, Bohr held science conferences to bring foreigners to his research institute. Behind the scenes these conferences were really job fairs in which Bohr assisted Jewish scientists to find sponsorship abroad. It was a time in which you were not permitted to leave the country without a foreign employer to work under.

One of the most interesting aspects of the lecture was what Bohr did when the Nazis took to the streets of Copenhagen. Bohr had been entrusted with the Nobel prizes of Max von Laue and James Franck who had remained in Germany. Their medals were successfully smuggled out of Germany at a time in which such exportations were considered to be capital crimes. The Nazis gathered any and all valuables to feed their war machine. The Nobel prizes remained at Bohr's institute for safekeeping, and as Professor Klein explained, Bohr began to worry considerably that the Nazis might take over the lab and find the medals. The recipient's names were engraved on them, and this would not have bode well for Laue and Franck had they fallen into enemy hands.

The thought of burying them was immediately ruled out for fear that they would be unearthed. George de Hevesy, a Nobel prize winner in chemistry, suggested that the medals be dissolved using a powerful acidic solution. They proceeded to precipitate the gold from acid and stored the medals in two separate unmarked jars. The Nazis ended up searching the lab and left the two jars containing the liquefied Nobel prizes alone. The jars were promptly sent to the royal mint in Stockholm to be recast as soon as the war was over [170]. It was a fascinating lecture and it was clear that this was a scientist's solution to a scientific problem.

You found any and all techniques that can be used to outsmart others fascinating, especially when it involved outsmarting evil tyranny. But how can this idea be extrapolated from the physical realm to the digital realm? You glance at the floppy drying next to you. How can we hide the Tetris Trojan from prying eyes? The way to do so is not clear at all. In the next instant a thought occurred to you. The salient aspect of the Bohr-Hevesy approach was that the gold was effectively melted to assume the same liquid form as the acid. The acid and gold were then intertwined at the atomic level, leading to an apparently worthless liquid. A separate process could later extract all of the Au atoms. This process could be repeated ad infinitum. How can a virus be seamlessly integrated into its host? One certainly cannot dissolve an assembly language virus. After all, this is the digital realm we are dealing with.

Given a high-level programming language J that can be decompiled, the solution is simple. Suppose that the host is written in J and suppose that the virus is written in J as well. The virus exists in compiled binary form, but totes around its J source code as well as a compiler and decompiler if needed. When the virus decides to infect a host, it decompiles the host. It then inserts its own viral source code into the host source code. The resulting infected source code is then compiled and saved, replacing the old program in the process. The virus ipso facto adheres to all of the compiler conventions of its host.[3] Depending on what compilers are available, the virus could be made to conform to the register and calling conventions of a gnu J compiler, a Microsoft J compiler, a Borland J compiler, and so forth. This would make the virus more difficult to detect. Of course there is ample room for improvement. It would be nice to be able to infer and subsequently mimic the high-level language programming *style* of the host program. You glance over at the floppy lying next to you. It is finally dry. This research topic will have to wait for another rainy day.

What the disk needs now is a new home. You pull open the top drawer and pull out a previously dismantled floppy. It had been prepared for just

[3]One could argue that the decompilability of Java is a security weakness that does not exist in the C++ language, for example. A language that behaves akin to a cryptographic one-way function during compilation guards against this vulnerability.

such an occasion. You set about reassembling the disk in its new housing. Moments later it is ready. You insert it into the drive in eager anticipation. The resident operating system mounts the floppy without a hitch. The file system properties have to be adjusted on the password file since the file was designated as invisible. You copy it onto your hard drive and eject the disk. After double-clicking on the file you find that it has 143 login/password pairs.

Presto.

When left to their own devices people choose the funniest passwords imaginable. This holds especially true for college undergraduates:

Login: pc541
Password: JoeIsUg1y

Login: sr412
Password: Pee$\emptyset$nMe

Login: ds912
Password: SnakeH$\emptyset\ell$e

You double-click on your saved collection and enter the password that is needed to decrypt it. One second later the plaintext file opens up in a text editor. You copy and paste the newly obtained passwords to your master list. Your running total is now 655. Some of the passwords were obtained via your password-snatching Trojan; still others were obtained from brute-force dictionary attacks. The university system administrators were still making the mistake of letting the Unix *passwd* file be easily accessible.

It was a good catch given that your last visit to the Trojan was only a week before. However, the running total is not really 655. You earmarked several of these accounts as potential honeypots. The most suspicious of all is:

Login: Password
Password: Password

Every time a user logs into a university machine, the user is warned that any unauthorized use is a criminal act and a violation of U.S. law. These honeypots are a way of trapping rogue users since they are easily guessed and grant access to accounts that are under 24-hour surveillance. You surmise that at least 620 of the user accounts should be safe to play with.

Before calling it a day you run your coin-flipping program. You type in 655 and let it flip away, prepared to toss the coin again if it winds up on a honeypot. The result comes up 422. You cross-reference this with your master list and determine that it's probably not a honeypot account.

Login: edc42
Password: m4Tds∅1

Tomorrow you shall be edc42 for a while.

* * * * *

The night is young as you step out of the house. In the distance, the Transco Tower stands silhouetted against the skyline. Like a municipal sentry it watches as the denizens of the city slowly make their way onto the streets. You smile in spite of yourself, for on this night you have something very special planned and you doubt that anyone has a vantage point good enough to see that the sequence of zeros and ones on the disk at your side is just a little unusual, a tad bit out of place, and in fact upon closer inspection downright insidious in nature.

You hop on your bike and pedal away from your home. Well, your home away from home is more aptly put. It has been over a year since you last visited your father, and the last time you were in Houston this new creation was little more than an idea on the drawing board. The city takes on a new hue as the headlights and 7-11 signs shed their evening glow. With not a cloud in the sky and no chance of rain, a new feeling begins to grow deep inside. This is the night. She will be free. You hope your due diligence will keep her alive. She will travel to strange lands and traverse hostile environments and will rely almost exclusively on what you taught her to do.

Turning a street corner you head out onto Westheimer. The traffic lights spread out as far as the eye can see, turning from red to green to yellow and back again in steady cadence. A small copy shop appears on the right. Looking inside you see a customer at a computer and a cashier looking off into space. Two people. No bustle. Not a chance. The last thing you need is a proprietor looking over your shoulder watching your every move. You pedal down an access road and jump a curb. Some more distance is necessary. Every mile counts. You traveled halfway across the country with her in tow, and it was important that you see her off safely.

A small shopping center comes into view. It is surprisingly full of cars. You swing into the lot and see a Burger King, a movie theater, a restaurant or two, and nestled in the middle of it all, an enticing-looking copy shop. Chaining your bike to the nearest pole, you scope the place out. Video cameras are mounted fore and aft, a convex mirror hangs over the computer area, there is no *uniformed* guard, and there is no tape measure along the side of the front door. Half a dozen customers are in line, and the copyists are buzzing around like worker bees. Drawing your attention to the computer area you see three people seated at computers, and another hovering over a color laser printer. It looks promising. In fact, it looks as promising as can be. Those *machines* don't stand a chance. You get up and walk into the establishment, just another face in the crowd, another customer needing to print out documents. You pass into the computer area and not a single person pays you notice.

As you seat yourself in front of a computer you try to recall how many times you washed the dishes after eating at a restaurant. Zero. Why? It's very simple really. It's not your responsibility. You pay for the food. You pay for the service. Cleaning up after yourself is *not your responsibility.* How silly would you look if you got up after paying, walked into the kitchen, and said "here, let me help you with those plates" to the employees? They would look at you sideways. Leaving your virus on this *general-purpose* computing machine is no different than this. The fact that others lack the cranial capacity to *see the grime* is not your fault. If it spoke to them the way it speaks to you then they'd be aware of her presence. But the privileged are few in number.

A message on the screen informs you of the rate: *Ten cents per minute.* The blinking caret asks you for your name. It is waiting. You give it one stochastically chosen from among the most frequent names in America: John. This machine is no longer for hire. It is temporarily yours, to do with as you will provided that it remains functional enough for the next customer. But most importantly, it must remain an intact vehicle for making the corporation money. That is what really matters in America. It will be so, you say to yourself, as you pull out the disk. It will be so.

You glance at the floppy in your hand and question once again your insatiable need to spread digital diseases. *This is the wrong thing to do.* Yet there is no helping your compulsion. You are diseased, but there is solace in the fact that your disease stems from the morally ruined society you live in. Greed begets punishment.

From idea in the back of your head, to scribbles on a drawing board, to mnemonics in an ASCII file, and on through the assembler your creation has traveled. You trained her on every antiviral program you could get your hands on. She bypasses them all. You click on the control panel and notice that the machine is running a virus shield. The shield consists of operating system hooks to file system interrupts that analyze their callers for suspicious behavior. Not a problem. Your virus already knows the location of the native interrupts needed to avoid the patches altogether. She has all the needed ROM addresses stored in her internal circularly linked list. You wonder how long it will be until she forgets them in lieu of newer ones, ones that have yet to be chosen by the computer manufacturer. You wonder if she will even live long enough to see that day. She is so clever, you think to yourself as you insert the disk, and fastidious too. She will never get a byte fatter than she already is.

A guy sitting down at the machine next to you looks over his printout. You make out a pie graph on it that seems to reference budgetary plans. While he is busy there making money you get busy running your infected version of Tetris. You spent well over a year designing her, calculating her cold-hearted offensive and defensive mechanisms.

...and now it is time to *bring down the machine.*

Seconds later a u-shaped block comes down the screen. Bingo. No system crashes and no antiviral warnings. She is free. She moved into her new home in the boot sector. But she has yet to leave her burrow and survey her surroundings. Experience has shown that crashes are not uncommon in these copy shops, so you hit the restart button knowing full well that the accounting software will not lose a second of billing time. As you eject the disk you look to the right and notice that the line has gotten even longer. The guy next to you signs out and heads to the back of the line to pay.

The machine boots up without a hitch. You see your name John on the start screen and click on the button labeled continue. Everything seems to be going smoothly. You take the liberty of running some of the resident programs: Photoshop, Microsoft Word, Acrobat reader, and a few text editors to boot. They all put on a few pounds. But who's going to notice? Infecting these programs manually is a good measure against any futile attempt to remove the virus. She's flying high, having beaten the heuristic scanners to the punch. She rerouted the interrupts first this time and will never be on the defensive on this particular machine *ever again.*

The clock in the lower right-hand corner of the screen reads "00:12." *Twelve minutes*. That's only one dollar and twenty cents. It was worth it. You sign out and head to the back of the line. A lady at the front of the line is unhappy with her glossy printouts. They are sprawled out all over the counter and she is taking forever to resolve the issue. The people behind her are visibly agitated.

You contemplate the future of your creation. Will she survive? Your gut tells you that she will. She has to. She has so much going for her. But how will you know how she fares in her new world? She'll never write. She'll never call. If you're lucky you might even hap upon her some day. Even then she won't tell you anything. You won't know how many children she has, or how many children her children have, and so on.

The lady reaches for her purse and pulls out a checkbook. Unbelievable. And you thought it was the information age. She finally takes her bag of glossies and leaves, to the relief of all. The line moves silently forward.

The problem with your creation is that she is an open book. Anyone can read her and with any luck she'll be notorious enough that lots of people will. You could have trained her to tell you of her exploits, but it is not clear how this could have been done in such a way that she would tell you and you alone. Ideally you would like one of her offspring to tell you some of the names of the infected machines and the order in which they were infected. You contemplated using Intel CPU IDs and IP addresses as unique identifiers to record the path that she and her progeny takes. However, one obvious problem with this approach is that the data could easily take up too much space in the virus. The real problem is the sheer stupidity of it. You may as well hand the Feds a road map to your house and draw a target on your back since it would have the same effect. Even encrypting the path with 3DES and storing the key within the virus would do little to hide the path from trained eyes.

A less detailed approach would be to divulge only the number of infected machines instead of the names of the machines and the order in which they were infected. This could be accomplished by giving her two generation counters. The idea is simple. A counter i would be included in the virus that would initially be set to a randomly chosen value r. Each time a virus has a child it increments i in the child by 1. For example, if the first virus had 8 offspring then they would each contain $i = r + 1$. Each of their children would have $i = r + 2$ and so forth. Another counter j would store the cumulative number of children that each forefather had.